



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

## PROGRAM PRO OPTIMALIZACI PROVOZU SERVISNÍHO STŘEDISKA

SERVICE CENTER OPTIMIZATION SOFTWARE

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Patrik Fillner

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Vladimír Levek, Ph.D.

BRNO 2020

# Diplomová práce

magisterský navazující studijní obor **Mikroelektronika**

Ústav mikroelektroniky

**Student:** Bc. Patrik Fillner

**ID:** 174291

**Ročník:** 2

**Akademický rok:** 2019/20

**NÁZEV TÉMATU:**

## Program pro optimalizaci provozu servisního střediska

### POKYNY PRO VYPRACOVÁNÍ:

Navrhněte program zpracovávající podklady servisního střediska, který umožňuje vyhodnocení a optimalizaci řízení lidských zdrojů ke zlepšení efektivity servisního střediska. Program bude sloužit k optimalizaci servisních úkonů ve společnosti Olympus Přerov. Program bude simulovat a optimalizovat paralelní a sériové servisní úkony a bude tím napomáhat ke snížení celkové doby servisu na jednom zařízení. Zároveň bude napomáhat ke snížení doby čekání jednotlivých komponent na servisní zásah a tím optimalizovat frontu operačních úkonů. Potřebné údaje a způsob vyhodnocení bude realizován dle intencí vedoucích servisního střediska.

V rámci diplomové práce popište možná řešení situace, vyberte vhodné řešení a to realizujte. Textovou část diplomové práce doplňte o potřebné výsledky vaší realizace a specifikujte efektivitu vaší práce jak z kvalitativního tak i z kvantitativního hlediska.

### DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 7.8.2020

**Vedoucí práce:** Ing. Vladimír Levek, Ph.D.

**doc. Ing. Lukáš Fojcik, Ph.D.**  
předseda oborové rady

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

## **ABSTRAKT**

Táto diplomová práca pojednáva o problematike optimalizácie servisného strediska a prináša návrh programu pre simuláciu a algoritmickú optimalizáciu. Zaoberá sa taktiež hierarchickým rozoberaním a logickou štruktúrou procesov. Taktiež sa zaoberá využívaním metód pre zisťovanie kvality a vizualizáciu procesov.

## **KLÍČOVÁ SLOVA**

Diagram príčin a následkov, Ganttov diagram, Histogram, Optimalizácia, Servisné stredisko

## **ABSTRACT**

This diploma thesis focuses on problematic about optimizing of service center and provides a base for creation of program for simulation and optimization. This diploma thesis also deals with hierarchical breakdown and logical structures of processes. Also there are quality determining methods and methods for visualization of processes.

## **KEYWORDS**

Cause and effect diagram, Gantt chart, Histogram, Optimization, Service center

FILLNER, Patrik. Program pro optimalizaci provozu servisního střediska. Brno, 2020. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/126939>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav mikroelektroniky. Vedoucí práce Vladimír Levek.

# PODĚKOVÁNÍ

Ďakujem vedúcemu diplomovej práce Ing. Vladimírovi Levkovi Ph.D. za účinnú metodickú, pedagogickú a odbornú pomoc a spoločnosti Olympus za poskytnutie údajov potrebných k tvorbe diplomovej práce.

# OBSAH

<b>Úvod</b>	<b>1</b>
<b>1 Teoretická časť</b>	<b>2</b>
1.1 Plytvanie .....	2
1.1.1 Definícia plytvania.....	2
1.1.2 Oblasti plytvania .....	3
1.2 Diagram príčin a následkov .....	3
1.2.1 Jednotlivé časti diagramu.....	4
1.2.2 Kategórie vplyvov.....	4
1.3 Hierarchický rozklad práce .....	5
1.3.1 Pravidlo 100 %.....	5
1.3.2 Výhody použitia hierarchického rozkladu .....	5
1.4 Ganttov diagram .....	6
1.4.1 Priradenie úloh a tvorba diagramu.....	6
1.5 Sieťový diagram.....	7
1.6 Histogram zdrojov .....	8
1.6.1 Návrh histogramu zdrojov .....	8
1.6.2 Tvar histogramu .....	9
1.7 Kľúčové ukazovatele výkonnosti .....	10
<b>2 Analýza problematiky</b>	<b>11</b>
2.1 Vymedzenie rozsahu problematiky .....	12
2.2 Analýza príčin a následkov .....	13
2.2.1 Získanie dát pre optimalizáciu programom .....	15
2.3 Hierarchický rozklad práce .....	16
2.3.1 Percentuálne rozdelenie úloh .....	16
2.4 Tvorba modelu siete.....	17
2.5 Posudzovanie kvality procesov a predikcia rizík.....	17
2.6 Návrh zobrazenia chodu systému .....	17
<b>3 Návrh a popis aplikácie</b>	<b>18</b>
3.1 Návrh užívateľského rozhrania .....	18

3.1.1	Jednotlivé časti aplikácie .....	18
	Modelovanie kostry strediska.....	19
	Zadávanie parametrov do modelu a optimalizácia.....	20
	Vykonávanie simulácie .....	22
	Reprezentácia a vyhodnocovanie výsledkov.....	22
3.1.2	Prvky pre návrh užívateľského rozhrania .....	23
	Prvok tabuľka .....	23
	Prvok stromová štruktúra .....	24
	Prvok textové pole.....	25
	Ostatné prvky .....	25
3.1.3	Ergonómia používania aplikácie.....	25
3.2	Návrh algoritmov .....	26
3.2.1	Univerzálnosť kódu .....	26
3.2.2	Testovanie a ošetrovanie stavov .....	27
3.2.3	Algoritmy pre ovládanie GUI.....	27
	Stromová štruktúra a načítanie údajov o technikoch .....	28
	Nastavenie úseku na mape modelu strediska .....	31
3.2.4	Kvantifikovanie liniek a úsekov .....	36
	Algoritmy pre výpočet intervalov .....	37
	Algoritmy pre výpočet váh.....	38
	Výpočet koeficientov odobratia a pridania technika.....	38
	Korekcia konštánt.....	39
3.2.5	Odhad času simulácie .....	40
	Algoritmy pre výpočet odhadu.....	41
	Porovnanie odhadov s reálnym priebehom .....	41
3.2.6	Automatizované priradovanie technikov .....	42
	Algoritmy rozvrhnutia technikov .....	43
3.2.7	Algoritmy pre optimalizáciu rozloženia technikov .....	45
	Hrubá optimalizácia pre rýchlosť prechodu.....	46
	Jemná optimalizácia pre rýchlosť prechodu.....	47
3.2.8	Simulácia chodu strediska .....	48
	Zjednodušený celkový pohľad na iteráciu simulácie .....	50
	Dátová štruktúra simulácie.....	52
	Presúvanie dokončeného prístroja.....	54

Priradzovanie prístrojov nevyťaženým technikom .....	56
3.2.9 Algoritmy pre vykresľovanie dát.....	58
<b>4 Testovanie aplikácie, skúšanie a overovanie</b>	<b>61</b>
4.1 Testovanie funkčnosti algoritmov .....	62
4.1.1 Prirad'ovacie algoritmy .....	62
4.1.2 Optimalizačné algoritmy.....	64
4.1.3 Simulácia prechodu jedného prístroja strediskom .....	66
4.1.4 Simulácia chodu strediska .....	68
Analýza prechodu prístrojov .....	69
Analýza vyt'aženia technikov .....	71
4.2 Overovanie funkčnosti aplikácie .....	73
4.2.1 Počiatočný model.....	73
4.2.2 Rozhodenie balansu strediska .....	76
4.2.3 Optimalizácia rozloženia .....	80
4.2.4 Zhrnutie overenia funkčnosti .....	82
<b>5 Závěr</b>	<b>83</b>
<b>Literatura</b>	<b>85</b>
<b>Seznam symbolů, veličin a zkratek</b>	<b>87</b>
<b>Seznam obrázků</b>	<b>88</b>
<b>Seznam grafů</b>	<b>90</b>
<b>Seznam tabulek</b>	<b>90</b>
<b>A Zdrojový kód dátovéj štruktúry simulácie</b>	<b>91</b>
<b>B Štatistika zdrojových kódov</b>	<b>93</b>
B.1 Jednoduchá štatistika súborov projektu .....	93
B.2 Stromová štruktúra projektu .....	94
B.3 Kiviat diagram súborov zdrojového kódu .....	95
B.4 Číselné štatistiky súborov zdrojového kódu .....	96
B.5 Číselné štatistiky metód zoradených podľa komplexnosti .....	97
<b>C Zoznam elektronických príloh</b>	<b>98</b>



# ÚVOD

Optimalizovanie procesov je jedným zo základných kameňov pre zvyšovanie kvality, ktorá je veľmi dôležitá a žiadaná naprieč celým priemyslom. Optimalizácia je však široký pojem. Môže mať veľa podôb a taktiež sa môže zameriavať na veľkú škálu rôznych parametrov. Samotná problematika servisného strediska je úzko špecifická a prináša radu rozdielov oproti bežnejším výrobným strediskám. Servisné stredisko pracuje so zariadeniami, ktoré sú rôzne a na vstupe tvoria pseudonáhodný zoznam druhov zariadení, podľa toho, ako prichádzajú zákazky do strediska. Jeho chod nie je možné jednoducho jednorazovo nastaviť a následne mierne upravovať, ale je potrebné neustále vykonávať zmeny napríklad v rozložení technikov a školeniach technikov, v závislosti na tom, aké zákazky zrovna prišli na vstup strediska. Výrobné stredisko oproti servisnému stredisku zvyčajne pracuje na sériových výrobách, kde je možné jednoduchšie naplánovať rozvrhnutie technikov, podľa toho, aká séria akých prístrojov sa aktuálne vyrába.

Diplomová práca sa zaoberá návrhom aplikácie, ktorá bude slúžiť, ako pomôcka pri optimalizácii servisného strediska. Práca na začiatok hľadá vhodné parametre, ktoré je potrebné a možné aplikáciou optimalizovať. Nachádzanie parametrov prebieha pomocou skúmania oblastí plytvania, ktoré popisuje problémy a pomôže nájsť ich symptómy. Následne práca tieto symptómy podrobí analýze príčin a následkov, ktorá pomôže určiť ciele vhodné pre optimalizáciu aplikáciou. Na základe týchto cieľov sú navrhnuté optimalizačné algoritmy.

K analýze chodu strediska s počiatočnými parametrami a analýze chodu strediska po optimalizácii neodmysliteľne patrí simulácia tohto chodu a vhodná reprezentácia výsledkov. Aplikácia teda musí obsahovať aj príslušnú simulačnú časť. Diplomová práca sa taktiež zaoberá aj hľadaním vhodných zobrazovacích metód a návrhom simulačných algoritmov. V práci sú všetky tieto algoritmy, rovnako ako aj chod aplikácie ako celku otestované a výsledky sú kvantifikované a vyhodnotené.

S témou optimalizácie nás oslovila spoločnosť Olympus, ktorá má v Českej republike servisné stredisko zdravotníckej techniky. Nadviazanie spolupráce medzi univerzitou a spoločnosťou prebehlo formou vypísania témy diplomovej práce. Túto tému som si vybral, pretože ma vždy zaujímala problematika simulácie, efektivity a optimalizácie. Programovaniu sa tiež aktívne venujem, práca pre mňa bola prínosom.

V diplomovej práci sú použité vlastné modely strediska, vhodné na zverejnenie. Mená technikov boli náhodne vygenerované z náhodného zoznamu mien a priezvisk. Výkonnosti technikov boli vygenerované procedurálne, podľa algoritmu pracujúceho s náhodnými číslami. Rovnako náhodne boli vygenerované aj názvy prístrojov a čísla zákaziek. Ukážky aplikácie v diplomovej práci neobsahujú citlivé údaje podliehajúce ochrane osobných údajov.

# 1 TEORETICKÁ ČASŤ

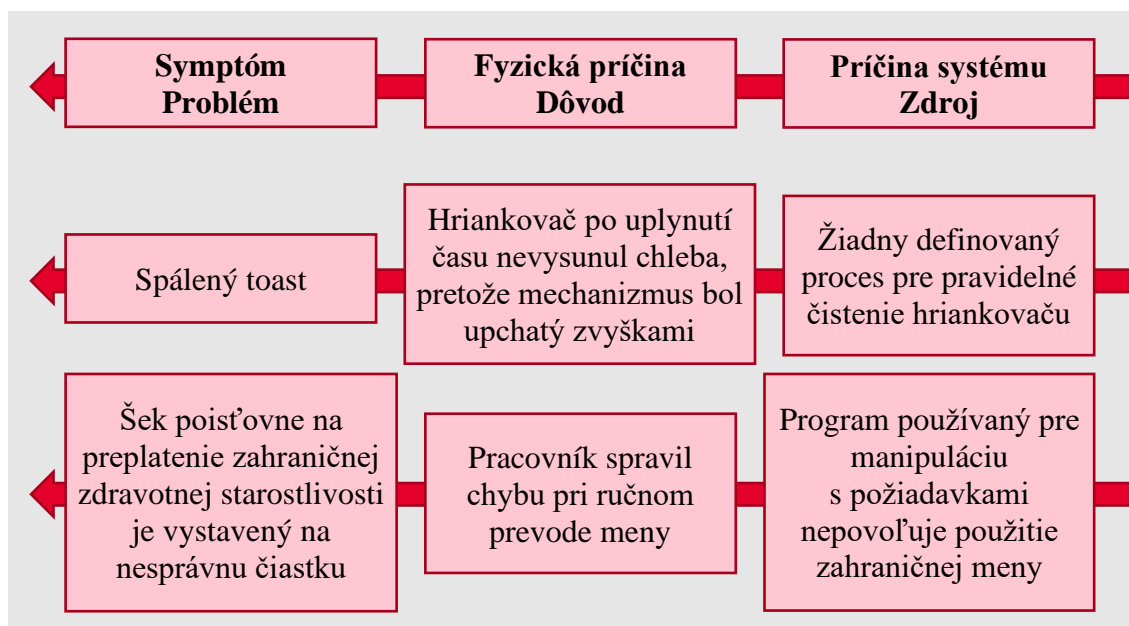
V teoretickej časti sú rozpísané rôzne metódy a nástroje, ktoré sú použité v práci, pre pochopenie a riešenie problematiky. Taktiež sú tu informácie o sledovaných faktoroch a parametroch používaných pre sledovanie a diagnostikovanie servisu.

## 1.1 Plytvanie

Základom pre optimalizovanie procesu je nájdenie slabých miest, ktoré by bolo dobré zlepšiť. Obzvlášť pri optimalizácii je nutné prehodnotiť efektivitu využitia prostriedkov a nájsť slabé miesta v podobe plytvania. Metodika *Lean Six Sigma* poskytuje zoznam oblastí pre identifikovanie plytvania, ktorý sa dá pre analýzu problému výhodné použiť. Tento nástroj je všeobecne uznávaný a široko používaný naprieč priemyselnými odvetviami pre zvyšovanie výnosov, pretože hlavným cieľom metodiky *Lean Six Sigma* je optimalizácia procesov pre dosiahnutie lepších výsledkov. Identifikácia a redukcia plytvania je efektívnou cestou k zvýšeniu profitu [1].

### 1.1.1 Definícia plytvania

Na elimináciu plytvania je nutné najskôr definovať plytvanie [2]. Je možné tvrdiť, že plytvanie je súbor nepotrebných aktivít [3], ktorých hlavné príčiny môžu byť stanovené analýzou oblastí plytvania [2]. Pre samotné vyriešenie plytvania je potrebné nájdenie zdrojov plytvania, pretože ak sa budú odstraňovať len symptómy hrozí, že staré problémy sa objavia v budúcnosti a aj vo väčšej mierke [4]. Pre lepšie pochopenie tejto teórie je na obrázku nižšie znázornené rozdelenie problému na symptómy a príčiny.



Obrázok 1 Rozdiely medzi symptómami a príčinami [5]

### 1.1.2 Oblasti plytvania

Pre zistenie príčin v už zabehnutých procesoch, ako bolo spomenuté v predchádzajúcej podkapitole, sa s výhodou dá použiť zoznam oblasti plytvania vychádzajúci z metodiky *Lean Six Sigma*. Jednotlivé oblasti plytvania je možné s príkladmi vyjadriť nasledujúcou tabuľkou uvedenou nižšie:

Tabuľka 1 Typy plytvania [3]

Typ plytvania	Príklad
Komplexnosť	Nepotrebné kroky, náročná dokumentácia, veľa povolení
Pracovná sila	Neefektívne operácie, nadbytočný počet zamestnancov
Nadprodukcia	Výroba väčšia, ako zákazník požaduje. Výroba pred tým, ako zákazník potrebuje.
Priestor	Priestor na zásoby, priestor pre súčasti čakajúce na využitie, prepracovanie, vyradenie a ostatné priestory
Energia	Plytvanie ľudskou energiou
Defekty	Opravy, prepracovanie, opakované úkony
Materiál	Vyhadzovanie, objednávky väčšie, ako potrebné
Nevyužitý materiál	Materiály, ktoré len čakajú v sklade
Čas	Plytvanie časom
Preprava	Pohyb, ktorý nemá žiadnu hodnotu
Bezpečnostné riziká	Nebezpečné prostredie, prípadne prostredie, ktoré môže byť základom pre vznik nehôd

Zoznam oblasti plytvania slúži ako prehľad najčastejších oblastí, v ktorých dochádza k určitej forme plytvania a z toho dôvodu je v každej literatúre možné nájsť jednotlivé oblasti, spolu s ich príkladmi mierne obmenené. Zoznam častokrát taktiež býva upravený, skrátený, alebo rozšírený v závislosti od konkrétneho použitia.

## 1.2 Diagram príčin a následkov

Tento diagram má v praxi používané rôzne názvy, napríklad *Ishikawa* diagram, alebo fishbone diagram (diagram rybej kosti) [6]. Používa sa na identifikovanie a organizovanie možných príčin problémov v štruktúrovanom formáte [6].

Tvorenie diagramu prebieha umiestnením problému na pravú stranu grafu, v pozícii pripomínajúcej „hlavu“ pomyslenej kostry diagramu. Hlavné príčiny sa kreslia ako vedľajšie „kosti“ hlavnej kostry diagramu [7].

Diagram je založený na pochopení variability vykazovanej procesom[8]. Variabilita v procese spôsobuje rôzne variácie výsledku procesu a práve zmieňovaný diagram dokáže prehľadne graficky zobrazit' zdroje variácie [8]. Pokiaľ však problém vyžaduje veľa úrovní príčin môže sa stať diagram neprehľadným, ťažkým na vytvorenie

a taktiež aj ťažkým na pochopenie [7]. Limitáciou diagramu príčin a následkov je možnosť zobrazenia iba jedného problému na jeden diagram [7]. Stále však slúži ako veľmi výhodný nástroj na vytvorenie podkladov pre zlepšovacie činnosti [8].

### 1.2.1 Jednotlivé časti diagramu

Diagram sa skladá z troch hlavných častí – hlava, chrbtová kosť a menšie kosti [6].

Hlava diagramu obsahuje popis problému, na ktorý sa zameriavame. Pozične je umiestnená na konci pravej strany diagramu [6].

Chrbtová kosť je hlavná kosť diagramu, dlhá, smeruje k hlave diagramu. Smer k hlave diagramu indikuje, že všetky veci, ktoré idu do chrbtovej kosti môžu spôsobiť problémy popísané v hlave diagramu [6].

Kosti diagramu majú rôzne veľkosti. Dlhé kosti napájajúce sa na chrbtovú kosť reprezentujú hlavné kategórie potenciálnych príčin. Jednotlivé menšie kosti reprezentujú hlbšie a hlbšie príčiny problému. Každá kosť je vlastne spojenie medzi príčinou a následkom, z čoho je poskladaný celý reťazec spájajúci najhlbšie príčiny s vybraným konečným problémom [6].

### 1.2.2 Kategórie vplyvov

Často sa k tvorbe grafov používajú štandardne používané kategórie vplyvov označované názvom prevzatým z angličtiny *5M and E*. Zoznam kategórií vplyvov je napísaný v nasledujúcej tabuľke:

Tabuľka 2 Kategórie vplyvov *5M and E* [9] [8]

<b>Materiál</b>	Fyzické veci, ktoré musia byť transformované, alebo využívané v priebehu procesu. Jedná sa o suroviny používané vo výrobnom procese, alebo pre iné procesy. Variácia materiálov vplýva na rôznorodosť vlastností produktov.
<b>Pracovná sila</b>	V procese sa vždy nájdu ľudia, ktorí sa na ňom nejakou činnosťou podieľajú a preto je treba uvážiť ich vplyv na chod procesu.
<b>Stroje</b>	Fyzické veci, ktoré sa počas procesu nespotrebovávajú, patria sem napríklad výrobné zariadenia, počítače, vozidlá a podobné veci. Variabilita pri použití strojov vzniká väčšinou pre neoptimálne vykonávanie postupov spôsobené opotrebovaním časti stroja.
<b>Metódy</b>	Operačné postupy, disciplíny a použité procedúry. Variabilita v metódach je obzvlášť značná, pokiaľ sú metódy založené prevažne na práci ľudí.
<b>Merania</b>	Spätná väzba použitá na kontrolu procesu.
<b>Prostredie</b>	Okolité a vonkajšie podmienky v ktorých sa proces vykonáva môžu mať značný vplyv na prevedenie procesu.

Zoznam kategórií vyššie je bežne a frekventovane používaný, avšak tieto kategórie len málokedy zahŕňajú všetky možné príčiny problému [7].

## 1.3 Hierarchický rozklad práce

Po definícii rozsahu projektu je ďalším krokom vytvoriť hierarchický rozklad práce [10]. V literatúre sa bežne používa skratka tvorená zo slov anglického názvu pre hierarchický rozklad práce *Work Breakdown Structure* – WBS. Rozklad práce pomáha rozdeliť prácu na projekte na menšie, lepšie ovládateľné kúsky [11].

Štruktúra sa navrhuje zostupne z definície cieľov projektu cez nasledujúce úrovne na najnižšie úrovne detailov vyžadované pre efektívny programový manažment [12].

Hierarchická štruktúra ponúka základ pre nasledujúce možnosti [12]:

- Definovanie práce potrebnej pre vykonanie v dostatočne veľkom detaile.
- Určuje nakoľko sú konečné súčasti príbuzné cieľom projektu.
- Identifikuje organizačné časti zodpovedné za úspešne zakončenie práce na každej nasledujúcej úrovni.
- Sumarizuje aktuálny stav a predpokladá vývoj projektu pre progresívne vyššie úrovne manažmentu.
- Vytvára detailné siete na požadovanej úrovni detailu.

Táto štruktúra zobrazuje hlavné prvky projektu, ako napríklad hardvér, rozhodnutia, služby, nástroje. Hlavné prvky sú potom rozdelené na jednotlivé komponenty, zahŕňajúce napríklad systémy, subsystémy, komponenty, ktoré sú opäť neskôr rozdelené na viac detailné jednotky [12].

### 1.3.1 Pravidlo 100 %

Najdôležitejšia podmienka pre návrh hierarchického rozkladu práce a vyhodnocovanie logiky rozkladu je pravidlo 100 % [13]. Definícia pravidla je nasledovná:

*„Rozloženie elementu v hierarchickom rozklade práce na ďalšiu úroveň musí reprezentovať 100 % práce potrebnej pre daný rozkladaný element“* [13]

Je dôležité si uvedomiť, že hlavný dôvod pre zavedenie použitia hierarchického rozkladu práce je z potreby zaistenia splnenia všetkých jednotlivých prác a aktivít a teda všetky práce a aktivity musia byť definované, naplánované a načasované [13]. Pravidlo 100 % taktiež napomáha pri tvorení hierarchického rozkladu práce k uvedomeniu si hĺbky vlastných znalostí o projekte [13].

### 1.3.2 Výhody použitia hierarchického rozkladu

Využitie hierarchického rozkladu prináša veľa výhod, jedná sa o celkom praktický a účinný nástroj. Niektoré z hlavných výhod sú uvedené v tejto podkapitole.

Použitie tohto nástroju prináša pochopenie práce na projekte už v počiatkovej fáze. Ihneď na začiatku sa totižto definuje rozsah projektu – produkty, služby a výsledky, čo má za následok odhalenie všetkých potrebných podúloh, ktoré môžu mať kritický vplyv napríklad na rozpočítanie nákladov, alebo vytvorenie časového plánu. Taktiež je možné dosiahnuť lepšieho porozumeniu procesom, aj v oblastiach s obmedzenými vedomosťami. [11].

Dobre navrhnutý hierarchický rozklad práce by mal taktiež obmedziť nutnosť nekontrolovaných zmien rozsahu projektu. Čím lepšie sa na začiatku navrhne

predpokladaný rozsah projektu, tým viac sa zníži pravdepodobnosť nutnosti náhlej zmeny rozsahu počas behu projektu. Zmeny rozsahu projektu počas behu projektu majú väčšinou za následok stratu kontroly nad projektom. Pri náhlej zmene rozsahu je však možné použiť pôvodnú štruktúru pre rýchlu implementáciu takejto zmeny. Prehľadná hierarchia môže slúžiť aj ako pomôcka pri schvaľovaní požiadaviek [11].

Hierarchický rozklad práce napomáha kontrole smerovaniu projektu a teda napomáha zabezpečiť dodanie očakávaných výsledkov. Dokáže vizualizovať internú a externú prácu. K smerovaniu taktiež patrí zaistenie zamerania pozornosti len pre veci, ktoré sú potrebné [11].

Tento druh plánovania ma ako ďalšiu z výhod minimalizovanie nutnosti preplánovania a pomáha skoro odhaľovať nástrahy, na ktoré je možné včas pripraviť preventívne opatrenia, alebo napláňovať korekčne akcie [11].

Veľká výhoda taktiež je, že hierarchický rozklad práce dokáže poskytnúť integráciu rozsahu projektu s časom a potrebnými finančnými prostriedkami [11].

## 1.4 Ganttov diagram

Ganttov diagram je efektívny nástroj pre plánovanie času, ktorý určuje aký časový úsek zaberie pre dokončenie úloh alebo aktivít [14]. Skladá sa z dvoch osí, na jednej má znázornené jednotlivé fázy a aktivity a druhá os je časová, ktorá obsahuje jednotky času v mierke hodín, dní, alebo týždňov [14]. V hlavnej oblasti grafu sú pre jednotlivé úlohy a aktivity nakreslené horizontálne čiary ukazujúce koľko času pre človeka, proces, alebo aktivitu je potreba na dokončenie úlohy [14].

V Ganttovom diagrame môžu byť taktiež znázornené aj rôzne príznaky, ktoré sa týkajú stavov, ako napríklad neprítomnosť operátora, nedostupnosť materiálu, oprava stroja, zmena alebo oprava nástroja [15]. Postup projektu je možné znázorňovať napríklad čiastočným zatmavením, alebo zatmavením celého horizontálneho pruhu [16].

Použitie Ganttovho diagramu pomáha zabezpečiť nasledujúce veci [16]:

- Všetky aktivity sú naplánované.
- Je započítané potrebné prevedenie (výkon, ľudia, prostriedky).
- Odhady času pre jednotlivé aktivity sú zapísané.
- Je vytvorený celkový odhad času potrebného na prevedenie projektu.

Výhodou tohto typu diagramu je prehľadnosť vizualizácie postupu a časovania projektu, avšak na úkor prehľadnosti zobrazenia postupnosti úloh [17]. Využitie Ganttovho diagramu pre zaistenie zdrojov a prostriedkov pre projekt je možné vypracovaním zoznamu požiadaviek na zdroje pre jednotlivé úlohy [18].

### 1.4.1 Priradenie úloh a tvorba diagramu

Po vyjasnení úloh potrebných pre splnenie projektu je potrebné tieto úlohy začať plánovať. Pre začiatok je dobré vytvoriť zoznam týchto úloh v sekvencií, v akej jednotlivé úlohy nasledujú [18]. Niektoré úlohy sa môžu prekrývať, prípadne môžu byť spracovávané paralelne [18].

Po vytvorení zoznamu je dôležité priradiť k úlohám ľudí, ktorí budú mať zodpovednosť za danú úlohu [18]. Toto priradenie má za úlohu zabezpečiť stiahnutie jednotlivých komponentov projektu v danom časovom úseku [18]. Pri priradovaní ľudí je možné rozdeliť priradených ľudí do troch kategórií [18]:

- Ľudia, ktorí majú schopnosti potrebné k uskutočneniu úlohy.
- Ľudia, ktorí majú záujem danú úlohu spraviť.
- Ľudia, ktorí si chcú zvýšiť schopnosti v danej oblasti spravením úlohy.

Hneď ako sú úlohy identifikované a zoradené v logickej postupnosti je treba k nim priradiť predpokladaný čas začiatku a predpokladaný čas ukončenia [18]. Tvorba samotného grafu je možná pomocou špecializovaného softvéru, ale je možné pre vytvorenie diagramu použiť aj obyčajný textový procesor, príklad grafu vytvoreného v textovom procesore je ukázaný nižšie:

Tabuľka 3 Príklad použitia textového procesoru na vytvorenie Ganttovho diagramu [18]

ID	Úloha	21. Apríla, 2019							28. Apríla, 2019							5. Mája, 2019			
		N	P	U	S	Š	P	S	N	P	U	S	Š	P	S	N	P	U	S
1	Míting																		
20	Normy				J														
21	Komunikácia				M	V													
22	Protokol								P										
23	Priradenia																		
24	Interné info.																		
Zodpovednosť: J – Ján M – Martin, V – Viktor, P - Peter																			

Na diagrame vyššie je možné vidieť jednotlivé prvky – úlohy, čas, horizontálne čiary úloh a jednotlivých priradených ľudí.

## 1.5 Sieťový diagram

Pre prípad komplexných projektov sa používa znázornenie pomocou sieťového diagramu. Obzvlášť, pokiaľ projekt obsahuje veľa inkrementálnych, alebo previazaných úloh, sa volí návrh zobrazenia pomocou sieťového diagramu, v ktorom jednotlivé vzťahy vidieť prehľadnejšie ako pri znázornení projektu pomocou Ganttovho diagramu, ktorý by mohol narásť do veľkých rozmerov [19].

Sieťový diagram k návrhu využíva najmä vzťahy medzi závislosťami úloh, ktoré bývajú nasledovné [20]:

- Koniec pre štart – v tomto vzťahu nasledujúca úloha nemôže začať skôr, ako skončí úloha predchádzajúca.
- Štart pre štart – tento vzťah hovorí o tom, že nasledujúca úloha môže začať spolu

so začiatkom predchádzajúcej a je možné, aby sa úlohy vykonávali paralelne. Oneskorenie štartu predchádzajúcej úlohy bude mať za následok oneskorenie úlohy, ktorá je závislá na jej štarte. Napríklad písanie používateľskej príručky môže začať spolu s definíciami požiadaviek, ale pokiaľ sa oneskorí definícia požiadaviek, nie je možné písať užívateľskú príručku.

- Koniec pre koniec – znamená, že nasledujúca úloha môže skončiť, len pokiaľ skončila aj predchádzajúca úloha. Napríklad pokiaľ je oneskorený užívateľský manuál, dodanie produktu na trh bude taktiež oneskorené.
- Štart pre koniec – málokedy používaný vzťah, ktorý značí, že ukončenie predchádzajúcej úlohy je viazané na začatie ďalšej úlohy, príklad by však mohol byť nasledujúci. Požiadavka pre technickú podporu nemôže byť ukončená, pokiaľ sa nezačne formovať tím pre riešenie technickej podpory.

Pri návrhu sieťových diagramov je možné vychádzať z hierarchického rozkladu práce spájaním jednotlivých elementov [21].

## 1.6 Histogram zdrojov

Histogram zdrojov je nástroj, ktorý zobrazuje vyťaženie zdrojov v každom časovom úseku projektu [22]. Histogram zdrojov je navrhnutý na zisťovanie nedostatočného, prípadne nadbytočného počtu prostriedkov [22].

Tento nástroj sa používa pri prispôbovaní využitia zdrojov. Prispôbovanie je založené na napláňovaní rovnomerného využívania zdrojov, či už sa jedná o zdroje kapitálu, nástrojov, vybavenia, alebo ľudské zdroje [23]. Bez napláňovania vyváženého vyťaženia zdrojov môže dochádzať k značnej fluktuácii požiadaviek na zdroje [23], čo môže viesť k oneskoreniu termínov [22].

V literatúre sa častokrát vyskytujú histogramy, ktoré majú rozdielne sledované parametre. Je to z dôvodu možnosti použitia dvoch rozdielnych základných veličín odvíjajúcich sa od dvoch rozdielnych pohľadov na využívanie zdrojov [23]. Tieto dva možné spôsoby využitia zdrojov majú prispôbovania nasledovné [23]:

- Časovo orientované prispôbovanie.
- Prostriedkovo orientované prispôbovanie.

Pre priame zobrazenie údajov o vyťažnosti a dostupnosti prostriedkov je možné k histogramu pridať vyčíslenie (bremeno) pre jednotlivé časové úseky zakladajúce sa na hodnotách predpokladu vyťaženia (údaj z histogramu) a dostupnosti prostriedkov (aktuálna dostupnosť prostriedkov v danom časovom úseku) [24].

### 1.6.1 Návrh histogramu zdrojov

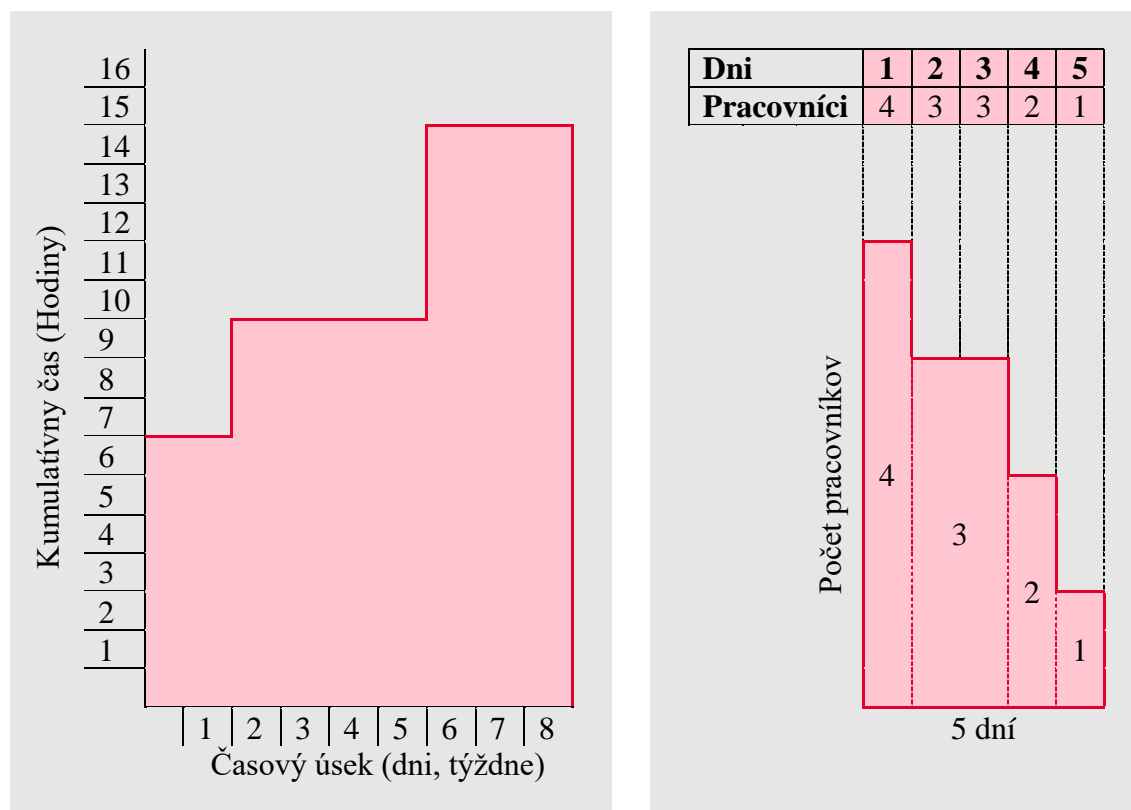
Telo histogramu pozostáva z dvoch na seba kolmých osí, ktoré slúžia ako kostra pre grafické zobrazenie dát [21]:

- Horizontálna osa vyjadruje časovú mierku v násobkoch, ktoré sú vyplývajúce z charakteristiky projektu.
- Vertikálna osa zobrazuje kumuláciu kvantifikácie zdroja a vyjadruje využitie prostriedku.
- Horizontálne pásma reprezentujú kumulatívne využitie zdroja počas každej



časovej jednotky na horizontálnej osi.

Príklad ilustrácie histogramov z dvoch rôznych zdrojov je zobrazený na obrázkoch nižšie:



Obrázok 2 Príklady histogramu pre časové [21] a prostriedkové [25] zobrazenie

Pri použití rovnakých časových jednotiek na horizontálnej osi je možné histogram využitia zdrojov integrovať k zobrazeniu rozvrhu aktivít pomocou Ganttovho diagramu a získať tak lepšiu prehľad o využívaní zdrojov jednotlivými aktivitami [23].

## 1.6.2 Tvar histogramu

Podľa typických tvarov histogramu môžeme posúdiť jednotlivé vlastnosti. Pokiaľ je profil histogramu nepravidelný môžu v projekte nastať potenciálne problémy [21]. Pokiaľ si je však manažér vedomý problémov, ktoré môžu nastať a zároveň sa s nimi dokáže vysporiadať, tak je nepravidelný tvar histogramu akceptovateľný [21].

Pokiaľ histogram obsahuje ostré špičky môžu v projekte nastať vážne problémy, ako napríklad nutnosť nadčasov v niektorých úsekoch projektu, čo môže mať za následok vyššie ceny práce a vyhorenie pracovníkov [21].

Priehlbiny v grafe majú opačný význam. Zamestnanci nemusia byť plne produktívny a môže dochádzať k dlhším periódam nevyužitého, prázdneho času, kedy sa môže napríklad stať, že budú dlho nečinne sedieť [21]. Nečinnosť pracovníkov väčšinou znižuje morálku, medzi nečinnými a produktívnymi zamestnancami môže dochádzať k interferenciám, produktívny zamestnanci môžu pociťovať neférové rozdelenie práce [21].

Nepravidelný tvar taktiež poukazuje na ostatné nefungujúce podmienky, napríklad

kontinuálne nabíranie a prepúšťanie členov tímu, čo im znemožňuje sústrediť sa na dôležité veci [21]. Nepravidelný tvar taktiež môže značiť aj iný potenciálny problém. Projektový manažér sa môže stať až príliš závislý na zamestnancoch s jedinečnými, špecializovanými vedomosťami [21]. Pokiaľ títo špecialisti odídu v špičke, ich odchod negatívne ovplyvní plán [21]. Pre zníženie tejto závislosti by projektoví manažéri mali získať viac ľudí so špecializovanými vedomosťami, krížovo vzdelávať ľudí a znížiť špičku (prispôsobenie využitia zdrojov) [21].

## 1.7 Kľúčové ukazovatele výkonnosti

Kľúčové ukazovatele výkonnosti sú všeobecne používaný pojem, ktorý je častejšie možné vidieť označený skratkou z anglického názvu *Key Performance Indicator* – KPI. Tieto ukazovatele reprezentujú sadu pre merania organizačnej výkonnosti, ktoré sú kritické pre súčasnú a budúcu výkonnosť organizácie [26]. Zaraďujú sa medzi tri kategórie metriky [27]:

- Indikátory výsledkov – „čoho bolo dosiahnuté?“.
- Výkonnostné indikátory – „Čo je potrebné zvýšiť na dosiahnutie potrebného výkonu?“.
- Kľúčové ukazovatele výkonnosti – „Ktoré výkonnostné ukazovatele sú kritické pre drastické zvýšenie výkonnosti, alebo uskutočnenie cieľov?“.

Dôležitý poznatok – manažment nie je len hra s číslami a kľúčové ukazovatele výkonnosti by primárne nemali byť len o meraniach [28]. Namiesto toho by mali slúžiť, ako hodnotné indikátory zamerané na plnenie stratégie [28]. Za každým ukazovateľom sa skrývajú ľudia, ako napríklad zákazníci, ktorí spravili objednávku, alebo zamestnanci a ich spokojnosť, alebo nespokojnosť [28].

## 2 ANALÝZA PROBLÉMATIKY

Práca cieli na vytvorenie aplikácie, ktorá bude slúžiť na optimalizáciu a simulovanie chodu servisného strediska. Problematika špecifických vlastností servisného strediska spočíva vo vykonávaní nejednotnej, nerovnomernej práce, závislej od prichádzajúcich zákaziek. Kontrastom k tejto charakteristike môže byť výrobné stredisko, ktoré sa vyznačuje sériovou výrobou. Sériová výroba zabezpečuje, že stredisko bude dlhodobo nastavené na jeden typ zariadenia a teda môžeme povedať, že výrobné stredisko má v porovnaní so servisným strediskom prácu pevne danú a jednoduchšie predvídateľnú.

Servisné stredisko obsahuje linky, ktoré sú poskladané z menších úsekov. Na úsekoch vykonáva technik operáciu. Operácia sa môže, ale nemusí skladať z viacerých krokov, ktoré môžu byť vykonávané bez prítomnosti technika, napríklad sušenie, alebo iné automatizované procesy.

Linky v stredisku rozdeľujú opravu na tri štádia. Príprava, oprava a dokončenie. Pre prípravu sa rovnako, ako aj pre dokončenie využíva len jedna linka, opravných liniek je v stredisku podľa druhu opráv viac. Prípravná linka sa spravidla končí rozobratím prístroja pred zahájením opravných prác a dokončovacia linka sa zas začína zložením všetkých kusov prístroja, ktoré prejdú opravnými linkami. Prístroj sa do opravných liniek dostáva podľa pravdepodobnosti vytvorenej zo štatistických údajov o poruchovosti konkrétnych častí konkrétnych prístrojov.

Zákazky vytvárajú zoznam prístrojov na vstupe do strediska. Stredisko taktiež môže obsahovať už rozpracované prístroje rozdelené na rôznych úsekoch. Operácie majú rôzne časy pre spracovanie prístrojov. Automatizované procesy majú časy pre každý typ prístroja pevne dané, procesy vykonávané technikmi majú čas závislý na výkonnosti obsluhujúceho technika. Každý prístroj má teda pre každú operáciu iný čas a ten sa zároveň ešte líši v závislosti na konkrétnom technikovi. Rýchlosti spracovania fronty prístrojov závisia od typov prístrojov vo fronte a taktiež aj od výkonnosti technika.

Technici majú okrem svojich časov na jednotlivých operáciách pre jednotlivé prístroje taktiež aj ďalší parameter a to ich schopnosti práce na úsekoch a prístrojoch. Každý technik je zaškolený len na niektoré úseky a iba pre niektoré prístroje. To znamená, že aj pokiaľ technik dokáže vykonávať nejakú operáciu na nejakom úseku, nemusí vedieť túto prácu na danom úseku vykonávať pre prístroje vo vstupnej fronte a v takom prípade je nutné nájsť technikovi úsek, na ktorom bude mať lepšie využitie vzhľadom na aktuálne zákazky. Niektorí technici môžu byť dokonca zaškolení len na jeden úsek, prípadne len na jeden prístroj. Technikov zaškolených len na jeden úsek je nutné na tomto úseku uprednostňovať a pokiaľ by boli vzhľadom na frontu zaradení nevyužití, poskytnúť užívateľovi aplikácie nástroj, ktorým by mohol zistiť prestoj technika a následne vykonať nápravu, napríklad zaškolením technika na iný druh prístroja z fronty, prípadne zaškoliť technika na iný úsek.

V tejto kapitole sa práca zaoberá praktickým využitím metód a nástrojov z predchádzajúcej teoretickej časti pre hlbšie analyzovanie problematiky a navrhuje využitie ukazovateľov procesných parametrov. Kapitola taktiež vymedzuje rozsah problematiky a slúži ako nosný podklad pre návrh programu.

## 2.1 Vymedzenie rozsahu problematiky

Na zistenie vhodných miest pre optimalizáciu bolo potrebné zo začiatku nájsť nástroj, ktorý by poukázal na kritické miesta a vymedzil rozsah, ktorým sa práca bude zaoberať. Pre spomínané požiadavky sa ukázala byť vhodná definícia z metodiky *Lean Six Sigma*.

V teoretickej časti je ukázaná celá definícia plytvania, v praktickej časti analyzujúcej problematiku sa rozbor plytvania vymedzuje na časti, ktoré sú vhodné pre optimalizáciu a o ktorých bude práca pojednávať.

Jednotlivé oblasti plytvania vyžadujúce optimalizáciu boli prebraté na stretnutiach so zadávateľom a sú nasledovné:

- Priestor            - Plytvanie priestorom pre výrobok čakajúci na ďalšie postupy.
- Energia           - Plytvanie ľudskou energiou, nedostatočné a nadmerné vyťaženie.
- Čas                - Plytvanie časom, zbytočné prestoje.

Plytvanie priestorom je značné pri neoptimalizovanom postupe jednotlivých procesov. Prejavy tohto druhu plytvania je možné vidieť hromadením spravovaných častí prístrojov na špecifických úsekoch. Toto môže mať za následok v krajnom prípade preplnenie priestoru čakajúcimi časťami prístrojov, čo môže viesť až k pozastaveniu činnosti predchádzajúceho pracoviska využívajúceho tieto priestory na prechodné odloženie výstupov.

Plytvanie ľudskou energiou sa spája s nedostatočným využitím potenciálu pracovníkov. Obzvlášť značný by mohol byť vyššie uvedený príklad, kedy v prípade preplnenia prechodných priestorov dochádza k pozastaveniu činnosti a tým pádom k nevyužitiu pracovníkov. Možným riešením sa zdá byť dočasné presunutie pracovníka na iný úsek, avšak pri dočasnom neplánovanom predelení pracovníkov na iný úsek, z dôvodu čakania na vytvorenie miesta pre výstupy nastávajú rôzne problémy. Pracovník na iný úsek nemusí mať školenie, pri príliš častých neplánovaných rotáciách medzi úsekmi sa znižuje pohodlie pracovníka, ktorý sa častým a hlavne neplánovaným striedaním rôznych prác môže prestať sústrediť na dôležité veci, čím sa zníži jeho výkonnosť a zvýši produkcia chýb v procesnom postupe.

Na druhú stranu k plytvaniu ľudskou energiou dochádza aj z druhej strany pohľadu na túto situáciu a to, z pohľadu pracovníkov na úseku, ktorý je zahltený čakajúcimi zariadeniami na vstupe. Zahltenie sa taktiež podpíše na komforte pracovníkov, na ktorých môže byť vyvíjaný tlak, prípadne sa z tejto situácie samí môžu cítiť byť pod tlakom a výsledok je podobný ako v predchádzajúcom prípade nevyťaženia. Dá sa očakávať zníženie výkonnosti spôsobené nepohodlím a hlavne zvýšenie produkcie chýb v procesnom postupe, ako dôsledok práce pod stresom. Konečná situácia môže priniesť výrazné zníženie celkového výkonu zvýšenou potrebou prepracovania chybného zariadenia zistenou až na výstupnej kontrole.

Plytvanie časom pozostáva zo zbytočnej nečinnosti. Typickým príkladom na najväčšie plytvanie je nečinné čakanie pracovníkov na úseku z dôvodu nedostatočnému počtu vstupov. Takáto nečinnosť má za následok neoptimalizovaný priechod opravovaných zariadení linkou. Do plytvania časom je však možné zahrnúť všetky udalosti tvoriace časové prestoje.

Z predchádzajúcej analýzy je možné vyhotoviť tabuľku symptómov jednotlivých oblastí plytvania. Tabuľka poskytne základný prvok – symptómy problémov pre ďalšiu analýzu a je uvedená nižšie (Tabuľka 4):

Tabuľka 4 Analýza symptómov

Druh plytvania	Problém	Symptóm
<b>Priestorom</b>	Preplnenie priestoru skladovaním zariadení čakajúcich na ďalší postup v procese.	Hromadenie čakajúcich zariadení
<b>Ľudskou energiou</b>	Niektoré úseky produkujú vyššie výstupy, ako zvládnu spracovať nasledujúce úseky v procese.	Väčšie výstupy medzi dvoma úsekmi, ako zvládnuteľné vstupy
<b>Časom</b>	Na niektorých úsekoch dochádza k zbytočnému čakaniu na dodanie vstupov pre spracovanie.	Menšie výstupy medzi dvoma úsekmi, ako zvládnuteľné vstupy

Symptómy sú najviac viditeľné prejavy nedostatkov. Pomocou nich je možné popísať problém, avšak nie je možné z nich priamo vyvodiť riešenie tohto problému. Z toho dôvodu je pre ďalšie spracovanie potrebné podrobiť symptómy ďalšej analýze, aby sa našli miesta, na ktorých zlepšenie sa aplikácia môže zamerať.

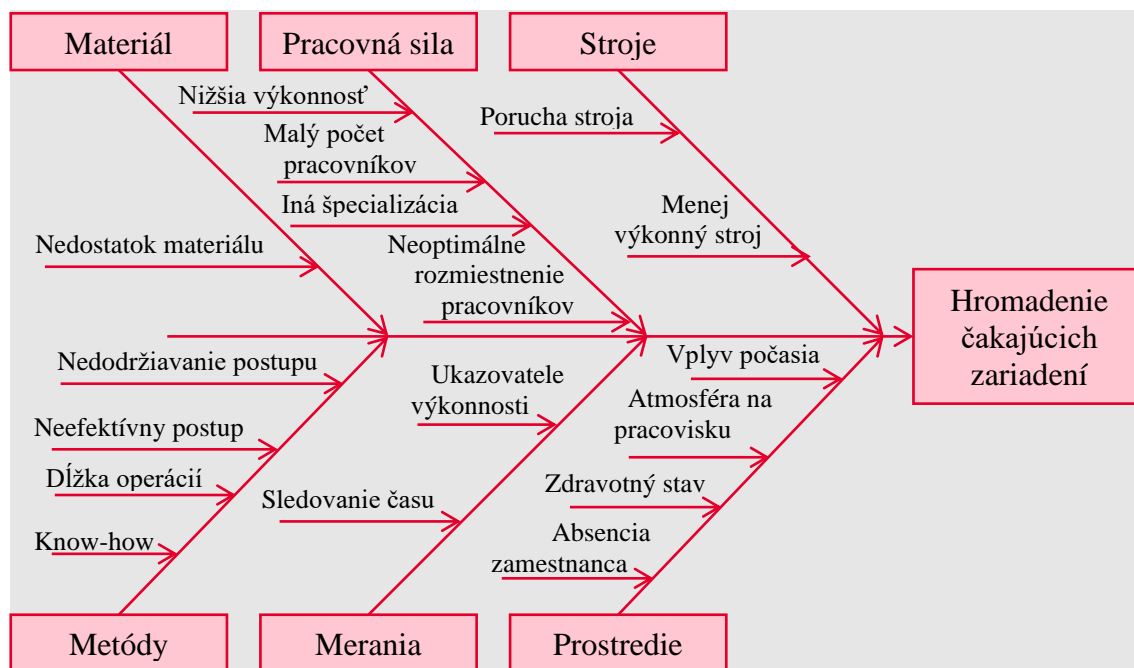
## 2.2 Analýza príčin a následkov

Časť vymedzenia problematiky našla miesta plytvania spolu so symptómami vychádzajúcimi z problémov týchto miest. Pre zlepšenie je však najskôr nutné nájsť a stanoviť parametre, ktoré dané plytvanie ovplyvňujú a s ktorými bude možné neskôr pracovať. Zoznám symptómov poskytuje do tejto analýzy následky, ku ktorým sú hľadané ich možné príčiny.

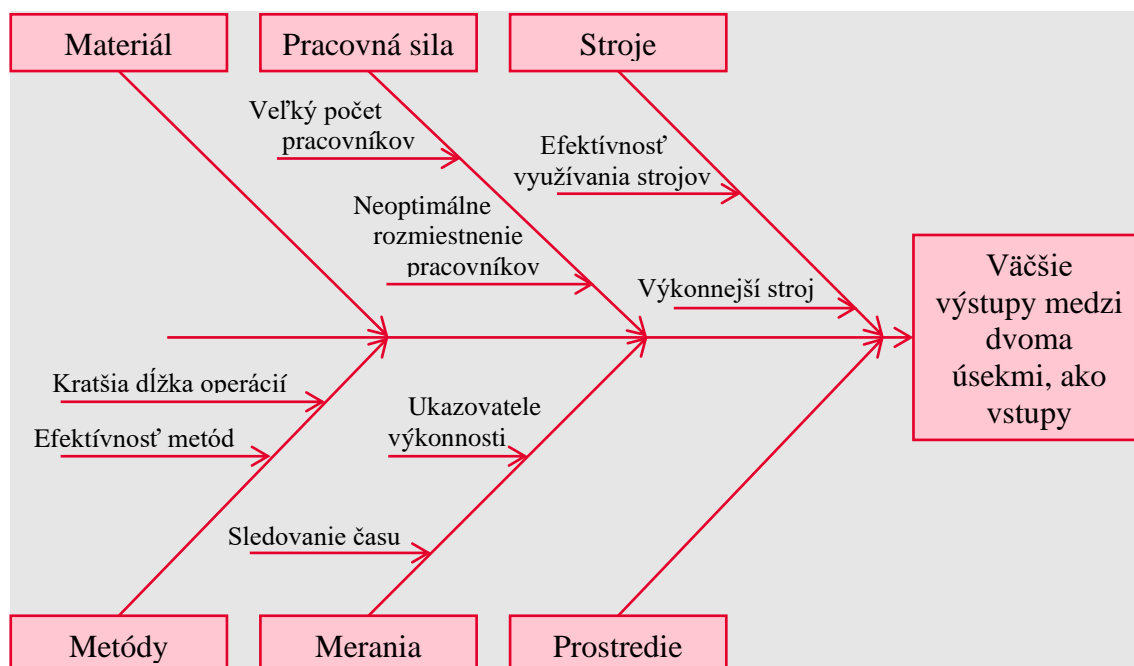
Prvým krokom je vytvorenie diagramu príčin a následkov. Po vypísaní všetkých možných príčin a parametrov, ktoré môžu zapríčiniť následok, je nutné vybrať pre konkrétny príklad plytvania len relevantné parametre. Niektoré parametre totižto nie je možné priamo ovplyvniť a teda nie je možné aplikáciou optimalizovať. Preto práve pre nájdenie cieľov optimalizácie je potrebné tieto parametre ešte hlbšie analyzovať.

Pri vytváraní diagramu a hľadaní príčin si je možné pomôcť pripravením kostry s kategóriami vplyvov z teoretického rozboru. Takouto analýzou aplikovanou na jednotlivé kategórie vplyvov je možné odhaliť hlbšie a hlbšie príčiny následku.

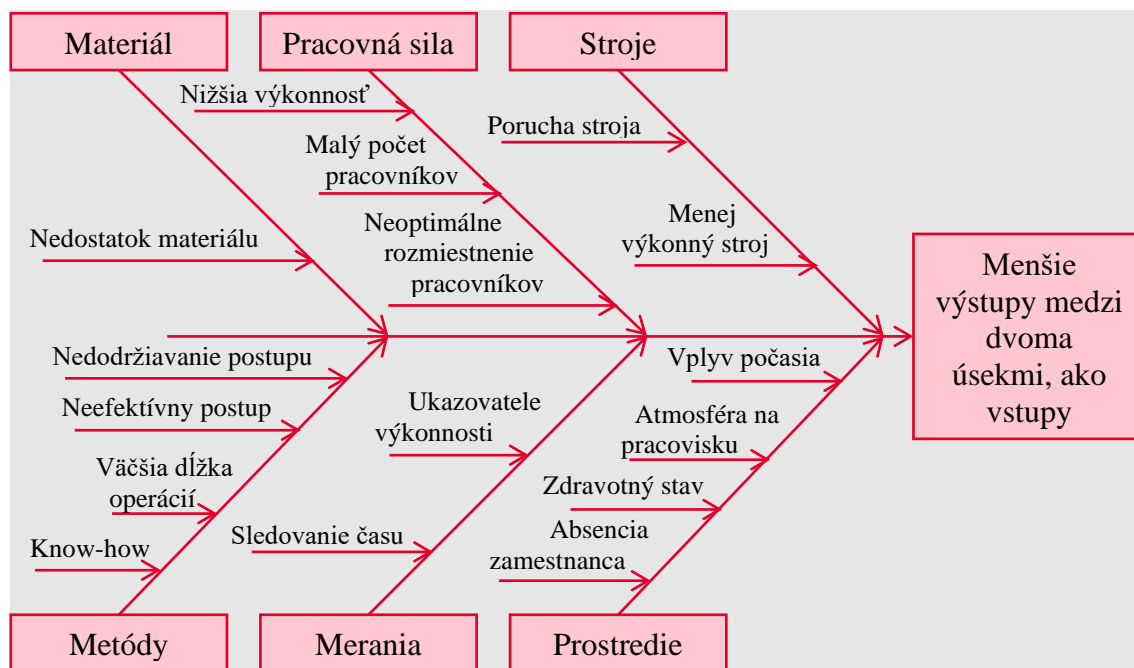
Diagramy príčin a následkov sú vytvorené pre každý symptóm a zobrazené sú na obrázkoch 4-6.



Obrázok 3 Diagram príčin a následkov pre plytvanie priestorom



Obrázok 4 Diagram príčin a následkov pre plytvanie ľudskou energiou



Obrázok 5 Diagram príčin a následkov pre plytvanie časom

Na uvedených diagramoch (Obrázok 3, Obrázok 4 a Obrázok 5) je možné vidieť širokú škálu príčin a parametrov pre každý symptóm. Diagramy je preto nutné ďalej analyzovať pre vytýčenie vhodných cieľov optimalizácie.

### 2.2.1 Získanie dát pre optimalizáciu programom

Vo vytvorených diagramoch je možné pozorovať, že jednotlivé príčiny rôznych druhov plytvania obsahujú spoločné znaky a teda sú si navzájom podobné. Nie však všetky príčiny z diagramov je možné optimalizovať systematicky programom, napríklad efektívnosť využitia strojov, alebo nedostatok materiálu. V diagramoch sú taktiež uvedené vplyvy, ktoré nie je možné ovplyvniť, ako napríklad vplyv počasia, alebo zdravotný stav.

Niektoré vplyvy taktiež programom optimalizovať nejdú, ale bolo by ich možné použiť ako upozornenie na možnú poruchu, napríklad pri porovnávaní dát simulácie s aktuálnymi výsledkami. Napríklad porucha stroja, zdravotný stav pracovníka alebo atmosféra na pracovisku. Pokiaľ by sa údaje o výkonnosti na jednom úseku začali výrazne odlišovať od simulácie, prípadne by ich odklon kolísal, alebo narastal, mohlo by to byť spôsobené problémom so strojom, prípadne zdravotným stavom alebo problematickou atmosférou na pracovisku. Pre presnejší odhad aktuálneho problému by bolo potrebné najskôr nájsť vzory reakcie systému na tieto problémy.

Pre prácu dôležité parametre sú parametre použiteľné k tvorbe programu pre simuláciu a optimalizovaniu chodu strediska. Jednotlivé parametre je možné rozdeliť na dva druhy parametrov. Prvým druhom sú parametre, ktoré popisujú stav a od ktorých sa bude odvíjať priebeh simulácie, ktoré sú unikátne a zo svojej podstaty nemenné. Druhý typ parametrov je možné jednoducho zmeniť a ovplyvniť pomocou návrhu optimalizovaného modelu simulácie. Vhodné parametre využiteľné pre návrh programu sú uvedené v tabuľke na ďalšej stránke (Tabuľka 4).

Tabuľka 5 Parametre vhodné pre použitie v programe získane analýzou príčin a následkov

Stavové parametre	Premenné parametre
Dĺžka operácie	Počet pracovníkov
Výkonnosť pracovníka	Rozmiestnenie pracovníkov
Absencia pracovníka	
Špecializácia pracovníka	

Parametre zobrazené v tabuľke vyššie sú základom pre simulačné a optimalizačné jadro programu. Toto jadro programu vykonáva základné funkcie celého konceptu a je podkladom pre pridávanie ďalších častí, ako sú rôzne ukazovatele, kľúčové ukazovatele výkonnosti, vizualizačné nástroje a iné pokročilé funkcie.

## 2.3 Hierarchický rozklad práce

Rozklad práce do určitej hierarchickej štruktúry je dôležitý nielen ako ukazovateľ časovej náročnosti, ale hlavne, ako nástroj, ktorý do chaotickej zmesi procesov vnáša hierarchickú štruktúru. V práci je využívaný pre rozbor pracovných postupov a ich vizualizáciu v programe. Tento typ vizualizácie napomáha zobrazit' percentuálne rozdelenie podprocesov v závislosti medzi potrebným časom na konkrétny podproces a celkovým časom potrebným k vyriešeniu samotného procesu.

Vizualizácia pomocou hierarchického rozkladu práce poskytne prehľad o pomernej časovej náročnosti jednotlivých úsekov strediska pre rôzne druhy servisovaných zariadení a zároveň v závislosti na konkrétnych pracovníkoch. Každý pracovník má inú výkonnosť a teda iný čas, za ktorý daný proces zvládne. Pre nájdenie adekvátnej náhrady v prípade absencie pracovníka je potrebné nájsť pracovníka, ktorý sa bude čo najviac svojou výkonnosťou v každej potrebnej oblasti / pre každé potrebné zariadenie v danom časovom úseku podobat' pracovníkovi s absenciou. Pomocou pomerového zobrazenia hierarchickým rozkladom práce bude možné prehľadne pozorovať pomerové (percentuálne) zmeny voči ostatným častiam procesov, ktoré táto náhrada prinesie.

### 2.3.1 Percentuálne rozdelenie úloh

Aby bolo možné určiť jednotlivé percentuálne rozdelenie úloh, je nutné najskôr určiť, koľko času trvajú jednotlivé úlohy a koľko času je 100 % času. Toto rozdelenie bude možné len pre konkrétny model založený na aktuálnom rozmiestnení pracovníkov. Samozrejme aktuálne rozmiestnenie pracovníkov môže byť brané ako počiatočná podmienka simulácie a tento model bude možné ďalej optimalizovať. Všeobecný model percentuálneho rozdelenia úloh pre konkrétne zariadenie by bolo možné vytvoriť z priemeru vykonávania jednotlivých úloh všetkými preškolenými pracovníkmi na dané úlohy na danom zariadení, však vzhľadom na individuálnosť časov by tento model mohol byť veľmi skreslený a vzdialený realite.



## 2.4 Tvorba modelu siete

Pre individuálne potreby simulácie procesu servisného strediska bolo treba nájsť vhodný systém, ktorý by vedel pracovať s veľkým množstvom kombinácií úloh, časov a ich závislostí na servisovaných zariadeniach. Každá z kombinácií zariadenie – úkon vyžaduje patričné školenie a keďže týchto kombinácií je vzhľadom na počet zariadení veľmi veľa, žiaden pracovník, aj keď na jednom rovnakom pracovisku nemôže opravovať každé zariadenie. V servisnom stredisku sa práca odvíja od aktuálnych individuálnych úloh pre veľa druhov rôznych zariadení, ktoré sa spracovávajú paralelne. Ako vhodný nástroj sa ukázal sieťový diagram, hlavne z dôvodu možnosti modularity. Pre tvorbu sieťového diagramu podľa aktuálnych požiadaviek je možné použiť definované časti siete pre konkrétne prípady.

Tvorba súčastí jednotlivých prvkov v sieti sa odvíja od potrieb jednotlivých procesov pre priechod konkrétneho zariadenia strediskom. Každý prvok v sieti musí obsahovať jednotlivé úlohy, ktoré sú preň potrebné a medzi úlohami musia existovať jasné logické závislosti.

Tvorba modelu siete pozostáva z logického pospájania jednotlivých prvkov, čo dokáže zabezpečiť jasnú logickú štruktúru popisujúcu potrebné závislosti medzi všetkými úlohami a podúlohami. S touto obrovskou logickou štruktúrou je možné ďalej pracovať a po priradení časov z aktuálneho rozmiestnenia pracovníkov hľadať vhodné postupnosti vykonávania pod-úloh (zariadenie – úkon) podľa logických vzťahov medzi nimi. Nájdením vhodnej postupnosti je možné proces optimalizovať a poskladať jednotlivé prvky tak, aby bolo rozloženie výkonu čo najviac rovnomerné a tým pádom sa znížilo ako hromadenie čakajúcich zariadení tak aj plytvanie časom.

## 2.5 Posudzovanie kvality procesov a predikcia rizík

Rozloženie práce je možné sledovať pomocou histogramu. Tento nástroj je jednoduché použiť a získať z neho veľmi cenné dáta predikcie. Pochopenie reprezentácie jednotlivých tvarov histogramu je rozpísané v teoretickom úvode, obecné, čím viac vyhladený histogram procesu bude, tým je na tom návrh procesu lepšie.

Z dát vygenerovaných simuláciou bude možné zostrojiť jednotlivé histogramy pre každú časť servisného strediska, ktorú bude možné nezávisle analyzovať.

## 2.6 Návrh zobrazenia chodu systému

Vizualizácia systému má základ v Ganttovom diagrame. Tento diagram bude obsahovať jednotlivé pracoviská a ich vytáženie v čase.

Ganttov diagram je efektívny nástroj na takýto druh zobrazenia, nie sú v ňom síce vidieť logické postupnosti, ale prehľadne zobrazí vrchnú vrstvu, v ktorej sú už len samotné pracoviská, pracovníci a zariadenia.

## 3 NÁVRH A POPIS APLIKÁCIE

Návrh aplikácie je možné rozdeliť na dve hlavné časti. Návrh užívateľského rozhrania a návrh samotnej logiky a algoritmov aplikácie. Užívateľské rozhranie poskytuje akýsi medzistupeň medzi užívateľom a samotnými algoritmami. Tieto dve časti je nutné robiť paralelne, aby sa docielilo čo najefektívnejšej implementácie požiadaviek a nedochádzalo k chybným interpretáciám.

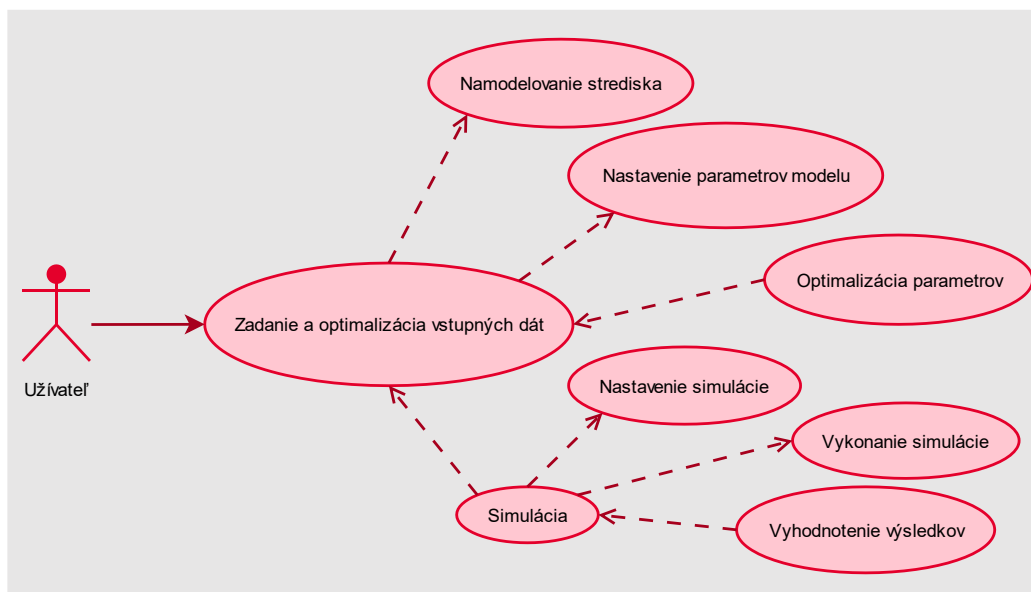
### 3.1 Návrh užívateľského rozhrania

Rozhranie by malo byť jednoduché a ľahko pochopiteľné, musí však obsahovať všetky príslušné prvky pre ovládanie funkcií. Pre uľahčenie ovládania užívateľom by malo byť ergonomické a intuitívne. Pre zaručenie funkčnosti aplikácie a správnosti výsledkov by malo mať ošetrené rôzne stavy, ktoré môžu nastať a ktoré môže užívateľ nastaviť, aj keď by nedávali zmysel. Pokiaľ nie sú ošetrené aj nezmyselné stavy, aplikácia môže spadnúť, prípadne môže dôjsť k chybnému výsledku.

#### 3.1.1 Jednotlivé časti aplikácie

Aplikácia slúži na simuláciu chodu strediska a optimalizáciu rozloženia technikov. Jednotlivé časti aplikácie teda logicky vychádzajú z týchto požiadaviek, ich vizualizáciu je možné vidieť na obrázku nižšie (Obrázok 6) a môžu návrh rozdeliť nasledovne:

- Časť pre namodelovanie strediska.
- Časť pre zadanie parametrov do modelu strediska a ich optimalizáciu.
- Časť nastavenia a vykonania samotnej simulácie.
- Časť pre reprezentáciu a vyhodnotenie výsledkov.



Obrázok 6 Diagram prípadov použitia pre návrh aplikácie, zjednodušená verzia

### Modelovanie kostry strediska

Hlavná kostra strediska sa skladá z jednotlivých úsekov v linkách. Každý jeden úsek má aspoň jedného technika, ktorý vykonáva na úseku rovnomennú operáciu. Typy všetkých liniek v stredisku rozdeľujú stredisko na tri časti.

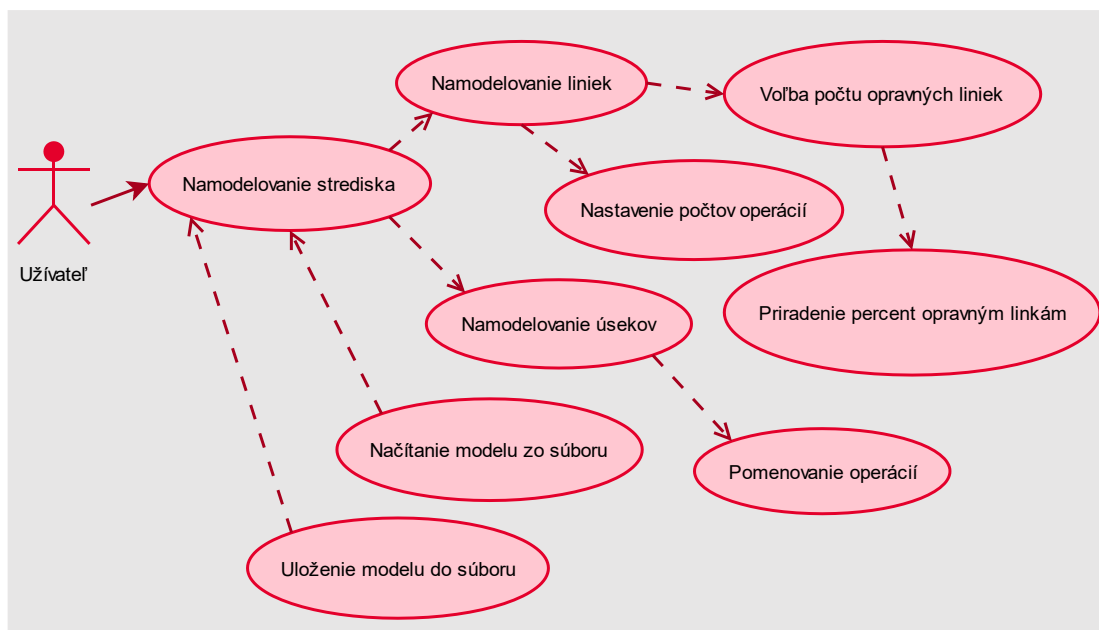
Prvá časť strediska obsahuje jednu linku, ktorá spracováva vstup výrobku do strediska. Obsahuje všetky prípravné operácie potrebné pre pripravenie prístroja do opravnej fázy, od prvej operácie vykonávanej po dodaní prístroja do vstupnej fronty na vstup strediska, až po rozobratie prístroja na jednotlivé časti pre ďalšie spracovanie.

Po rozobratí sa prechod rozvetvuje do viacerých liniek pre opravu. Do každej linky štatisticky vstupuje určité percento častí z celkového počtu prístrojov po rozobratí. Pre lepšie pochopenie problematiky uvediem pár príkladov. Napríklad pri rozobratí prístroja môže dôjsť k netesnostiam v gumových tesneniach krytu a z tohto dôvodu je potrebné pre každý rozobratý prístroj spraviť výmenu tesnení krytu. Linkou určenou na výmenu tesnení krytu teda musí prejsť presne toľko kusov, koľko kusov prístrojov prejde rozobratím. Linkou teda prejde 100 % rozobratých prístrojov. Menší počet percent prejdencých prístrojov vychádza zas zo štatistických údajov o poruchovosti prístrojov, napríklad najčastejšou chybou môže byť zlomenie namáhanej súčiastky, ktoré by mohlo byť príčinou nutnosti servisného zásahu až pre 60 % zariadení. Touto linkou teda štatisticky prejde 60 % z celkového počtu rozobratých prístrojov. Percentá liniek v oprave neurčujú percentuálny pomer, ale percentuálny príjem prístrojov z celkového počtu prístrojov prejdencých úsekom pre rozobratie, podľa pravdepodobnosti potreby daných servisných úkonov. Z tohoto dôvodu neplatí, že by sa súčet percent v linkách opravy mal rovnať 100 %.

V poslednej časti dochádza ku skladaniu a testovaniu zloženého prístroja. Skladanie prístroja a teda možnosť spracovania prístroja prvým úsekom linky prebieha až po príchode všetkých častí prístroja z opravných liniek na vstup tejto linky. Táto časť rovnako ako prvá časť obsahuje len jednu linku. Linka pokrýva operácie od zloženia prístroja až po poslednú operáciu v stredisku.

Z tohto rozdelenia je možné usúdiť, že modelovanie kostry strediska vyžaduje grafické rozhranie umožňujúce zvoliť počty úsekov v jednotlivých linkách a taktiež umožňujúce priradiť názvov operácie vykonávanej úsekom. V časti opráv prístrojov je navyše potrebné v grafickom rozhraní umožniť taktiež aj nastavenie počtu liniek a ich vyššie spomínaný percentuálny príjem.

Na základe týchto požiadaviek môžeme vytvoriť diagram prípadov použitia pre namodelovanie strediska. Diagram je uvedený na obrázku nižšie (Obrázok 7). Rozširujúce prvky zobrazené v diagrame pre prácu so súbormi súvisia s ergonómiou používania aplikácie, ktorá je v práci rozobratá v časti 3.1.3 Ergonómia používania aplikácie. Ukladanie modelu kostry strediska musí zabezpečiť získanie hodnôt z grafického rozhrania, ich následné spracovanie pre uloženie do súboru. Načítanie musí obsahovať blok pre preloženie dát načítaných zo súboru, upravenie grafických prvkov pre načítané dáta a naplnenie týchto prvkov údajmi.



Obrázok 7 Diagram prípadov použitia pre namodelovanie kostry strediska

### Zadávanie parametrov do modelu a optimalizácia

Pre upresnenie parametrov je v prvom rade potrebné definovať, ktoré parametre sú potrebné pre chod strediska. Ďalším krokom k návrhu je definovanie funkcií, ktoré by mohli byť vyžadované pre obsluhu aplikácie, pre poskytnutie komfortu používania, a najmä pre zníženie zavedenia chyby do simulácie.

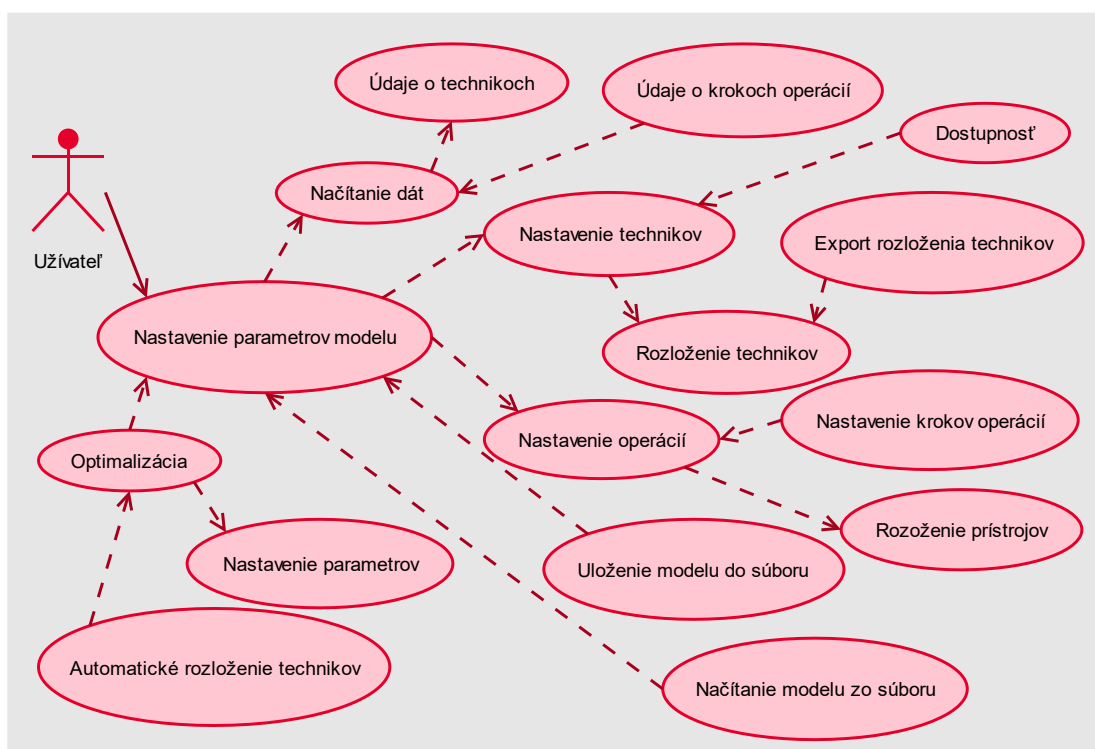
Parametre ovplyvňujúce chod strediska a prechod prístrojov sú:

1. Údaje o výkonnosti technikov a znalosti práce na konkrétnom úseku, čo znamená schopnosti technikov pracovať na jednotlivých úsekoch a ich výkony pre tieto úseky v závislosti na konkrétnom spravovanom prístroji.
2. Údaje o schopnostiach technikov spracovávať frontu zariadení, s čím sú spojené informácie o schopnostiach technikov spracovávať prístroje na danej operácii.
3. Rozvrhnutie technikov – automatické rozmiestnenie technikov na základe ich schopností uvedených v bodoch vyššie, prípadne manuálne rozmiestnenie technikov podľa skutočných stavov strediska a s tým spojené ošetrenie, aby nemohol byť technik priradený na úsek, na ktorom nevie vykonávať prácu.
4. Rozvrhnutie prístrojov – možnosť vytvorenia fronty čakajúcich prístrojov na konkrétnych úsekoch, či už fronty prístrojov na začiatku strediska a teda frontu nových zadaných zákaziek čakajúcich na vstupe, alebo fronty na jednotlivých úsekoch podľa aktuálnych stavov rozpracovaných zákaziek.
5. Modely samostatných krokov v operáciách – namodelovanie jednotlivých operácií na menšie kroky, ktoré sú zložené z krokov, ktoré obsluhuje technik a krokov, ktoré sa vykonávajú bez prítomnosti technika, napríklad sa jedná o temperovanie, sušenie, alebo sú to plne automatizované činnosti, ktoré sa vykonávajú bez prítomnosti technika. Toto modelovanie menších krokov je dôležité v celkovej simulácii chodu prístrojov, pretože počas času prístroja v kroku nezávislom na technikovi technik buď čaká, alebo začína spracovávať ďalší prístroj. Jednotlivé kroky je možné zjednodušiť, ale čím presnejšie budú

operácie namodelované, tým presnejší bude priebeh simulácie.

6. Údaje o časoch pre jednotlivé kroky, ktoré nie sú závislé na technikovi, pre každý prístroj – pre doplnenie údajov do modelov.

Z vyššie spomínaných parametrov vyplýva, že optimalizácia môže prebiehať jedine zmenou rozmiestnenia technikov. Parametre, ako aktuálne rozloženie prístrojov čakajúcich na spracovanie, výkonnosti technikov, modely krokov a ostatné zmienené parametre majú síce taktiež vplyv na chod strediska, sú však pevne daná a nie je ich možné meniť. Pre optimalizáciu je možné vymyslieť viacero algoritmov. Z hľadiska jednoduchosti obsluhy je dobré, aby okrem optimalizácie už priradených technikoch na úsekoch aplikácia taktiež obsahovala funkciu pre automatické optimalizované priradzovanie nepriradených technikov. Pokiaľ je technikov v stredisku veľa, výrazne to zvýši komfort používania aplikácie. Potrebne prvky sú uvedené v zjednodušenej verzii diagramu prípadov použitia nižšie (Obrázok 8).



Obrázok 8 Diagram prípadov použitia pre nastavenie a optimalizáciu parametrov modelu, zjednodušená verzia

Medzi funkcie pre uľahčenie zadávania parametrov do modelu môžeme zaradiť napríklad možnosť nastavenia dostupnosti technikov. Nastavenie dostupnosti môže byť realizované pridaním možnosti jednoducho označiť parameter vyjadrujúci dostupnosť, prípadne nedostupnosť jednotlivých technikov, či už z dôvodu choroby alebo dovolenky a tomu patrične prispôbiť priradovacie a optimalizačné algoritmy. Technikovi sa v prípade nedostupnosti samozrejme dá ručne odstrániť a v prípade opätovnej dostupnosti zas pridať záznam, táto funkcia však zjednodušuje ovládanie. Ďalšou funkciou, ktorou je možné zjednodušiť ovládanie je napríklad automatické skontrolovanie, či sa v prípade manuálneho priradenia nejakého technika na úsek daný technik v stredisku už nenachádza. V prípade, že sa technik nachádza na inom úseku,

namiesto obvyčajného priradenia technika tohto technika presuň. Implementovaním tejto funkcie odpadá nutnosť manuálnej kontroly a odstraňovania duplicitných priradení technikov. Dochádza k zvýšeniu ergonomie aplikácie a zníženiu možnosti zavedenia chyby ľudským faktorom – nepozornosťou obsluhy aplikácie.

Ďalšími drobnými, ale užitočnými funkciami pre uľahčenie používania aplikácie môžu byť napríklad prehľadné zobrazenie zoznamu technikov pre jednotlivé operácie a exportovanie tejto informácie do súboru použiteľného aj mimo aplikácie – napríklad do textového dokumentu.

Všetky hlavné funkcie pre uľahčenie ovládania a používania aplikácie uvedené v odstavcoch vyššie sú v značne zjednodušenej forme taktiež uvedené na obrázku (Obrázok 8).

### **Vykonávanie simulácie**

Popri vykonávaní simulácie nie sú potrebné žiadne dodatočné zásahy užívateľa do chodu aplikácie a z toho dôvodu môže byť toto grafické rozhranie jednoduchšie.

Pred vykonaním simulácie je však potrebné vedieť počet hodín, ktorý sa ma simulácia vykonávať a teda je potreba aspoň dvoch prvkov, jeden pre zadanie dĺžky simulácie a druhý pre potvrdenie spustenia simulácie. Ďalšie dodatočne parametre môžu byť napríklad v prípade potreby rôzne scenáre rozhodovacej logiky simulácie. V prípade možnosti uloženia vykonanej simulácie a možnosti vykonania viacerých simulácií je taktiež vhodné pridať parameter názov simulácie na jednoduché spätné rozoznanie uloženej simulácie.

Počas vykonávania simulácie je dobré užívateľovi ukázať, v akom štádiu sa simulácia aktuálne nachádza a ako rýchlo sa simulovanie blíži ku koncu. Pre tento účel je výhodne použiť vhodný ukazovateľ priebehu.

Po vykonaní simulácie je dobré zobrazit' stavy strediska po prebehnutí simulácie. Užívateľ aplikácie môže rýchlo skontrolovať priebeh simulácie a pomocou porovnania s počiatočnými stavmi strediska posúdiť efektivitu rozloženia technikov v procese pre konkrétnu simuláciu. Rýchly prehľad stavov umožní efektívnejší zásah do modelov simulácie, ešte pred podrobným rozborom jej výsledkov.

### **Reprezentácia a vyhodnocovanie výsledkov**

Dátová štruktúra simulácie obsahuje veľa informácií, ktoré sa dajú výhodne použiť na podrobné vyhodnocovanie chodu strediska, je nutné im však priradiť určitú vhodnú formu. Vychádzajúc z teórie môžeme pre reprezentáciu využitia prostriedkov použiť napríklad príslušný histogram. Pre reprezentáciu chodu strediska v čase môžeme zas použiť Ganttov diagram. Pre vyhodnotenie kvantitatívnych výsledkov sú vhodné číselne ukazovatele, ktoré vhodným výpočtom môžu vyjadrovať napríklad počet spracovaných kusov, počet rozpracovaných kusov v stredisku, dĺžku prechodu prístroja a iné kľúčové ukazovatele výkonnosti.

Ďalšie dodatočné spracovanie dát môže byť prevedené pomocou externej aplikácie, podľa užívateľovho výberu. Pre túto možnosť je nutné, aby aplikácia taktiež dokázala potrebné dáta exportovať do nejakého univerzálneho formátu, napríklad do csv súboru. Pomocou tohto typu súboru má užívateľ možnosť využiť vlastnú reprezentáciu dát.

### 3.1.2 Prvky pre návrh užívateľského rozhrania

Pre návrh grafického užívateľského rozhrania bola použitá sada nástrojov grafických komponent *wxWidgets*. Táto sada nástrojov obsahuje rôzne prvky, z ktorých je možné efektívne poskladať grafickú aplikáciu.

#### Prvok tabuľka

Prvok tabuľka je zložený z buniek obsahujúcich dáta a z buniek obsahujúcich popisy riadkov a stĺpcov. Tieto popisy riadkov a stĺpcov je možné skryť a tabuľku tým pádom použiť napríklad na vykreslenie modelu strediska (Obrázok 9), prípadne procesného modelu (Obrázok 10).

Vstupna kontrola	Diagnostika	Rozobratie	Vymena tesneni	Mazanie	
			Diagnostika dosky	Vybratie dosky	Vymena dosky
			Odstranenie konektoru	Vymena konektoru	Kontrola konektoru

Obrázok 9 Ukážka kúsku tabuľky so skrytými popismi využitej na vykreslenie mapy modelu

--- Linka 0 ---	Vstupna kontrola	Diagnostika	Rozobratie			
--- Linka 1 ---	Vymena tesneni	Mazanie	temperovanie	Mazanie	temperovanie2	Mazanie
--- Linka 2 ---	Diagnostika dosky	Vybratie dosky	Vymena dosky	Testovanie dosky		
--- Linka 3 ---	Odstranenie konektoru	Vymena konektoru	Kontrola konektoru	automatizovana skuska pevnosti		
--- Linka 4 ---	Zlozenie	Opt+El kontrola	Funkcna kontrola	Vystupna kontrola		

Obrázok 10 Využitie tabuľky so skrytými popismi stĺpcov pre mapu procesného modelu

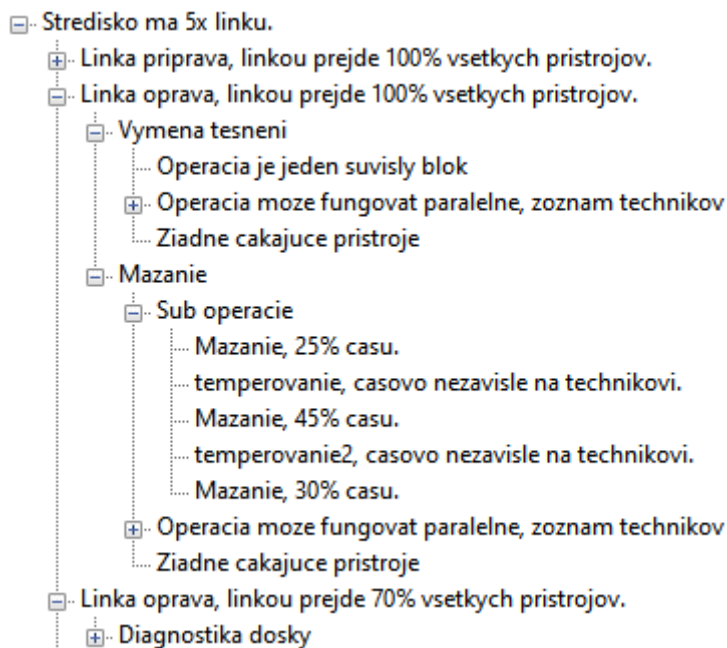
Bunkám v tabuľke je taktiež možné priradovať rôzne podfarbenia, čo sa dá s výhodou využiť na vykresľovanie Ganttovho diagramu. Ukážka vykreslenia Ganttovho diagramu je na obrázku nižšie:

	0:00	0:15	0:30	0:45	1:00	1:15	1:30	1:45	2:00	2:15	2:30	2:45	3:00	3:15	3:30	3:45	4:00
--- Linka 0 ---																	
Vstupna kontrola																	
Diagnostika																	
Rozobratie																	
-																	
--- Linka 1 ---																	
Vymena tesneni																	
Mazanie																	
-																	
--- Linka 2 ---																	
Diagnostika dosky																	
Vybratie dosky																	
Vymena dosky																	
Testovanie dosky																	

Obrázok 11 Ukážka využitia tabuľky pre vykreslenie Ganttovho diagramu

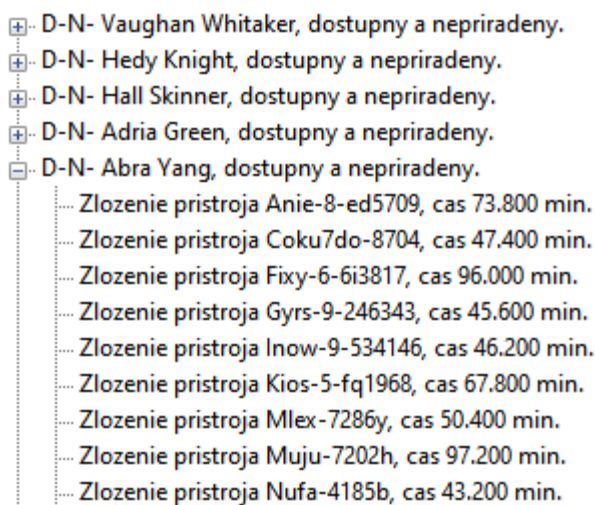
### Prvok stromová štruktúra

Užitočným prvkom pre zobrazenie veľkého množstva štruktúrovaných informácií je stromová štruktúra. Zapisovanie do stromovej štruktúry prebieha pomocou zapísania informácií na hlavný koreňový uzol, z ktorého sa záznamy môžu ďalej pomocou ďalších uzlov rozvetvovať. Jednotlivé uzly sa dajú jedným kliknutím rozvinúť a ďalším kliknutím zas zabaliť. Stromový zápis (Obrázok 12) preto umožňuje rýchle hľadanie konkrétneho prvku v štruktúre, aj pokiaľ je štruktúra značne rozvetvená.



Obrázok 12 Ukážka využitia stromovej štruktúry

Pre štruktúrované dáta, ktorých koreňový údaj nemá značnú informatívnu úlohu je možné tento koreňový uzol skryť. Parameter skrytia sa nastavuje pri samotnom návrhu. Skrytie koreňového uzlu (Obrázok 13) v takejto situácii sprehl'adní zápis.



Obrázok 13 Ukážka stromovej štruktúry so skrytým koreňom



Algoritmom pre zapisovanie dát do stromovej štruktúry sa podrobnejšie venuje podkapitola *Stromová štruktúra a načítanie údajov*. Ilustrácia takeého algoritmu je uvedená v danej podkapitole na obrázku so zjednodušeným diagramom aktivít pre načítanie technikov do stromovej štruktúry (Obrázok 14).

### **Prvok textové pole**

Ďalším dôležitým prvkom pre návrh užívateľského rozhrania je textové pole. Pre zadávanie textových údajov sa jedná o nevyhnutný komponent, ale dá sa s výhodou použiť aj pre zobrazovanie výstupných textových údajov. Použitá sada nástrojov grafických komponent ponúka na výber rôzne druhy textových polí a k nim veľkú škálu rôznych parametrov. Polia môžu byť jednoriadkové, viacriadkové, so zalamovaním riadkov, bez zalamovania riadkov, jednoduché s obyčajným textom, s možnosťou pokročilých textových efektov a podobne. Za zmienku taktiež stojí možnosť zmeniť v každom textovom poli font. Táto možnosť umožní použiť napríklad písmo s pevnou šírkou pre vykreslenie textovej grafiky.

Výhodou použitia textového poľa pre výpis výstupných údajov je možnosť skopírovania textového obsahu pre ďalšie použitie mimo aplikácie, napríklad vytlačenie obsahu, zaslanie pomocou e-mailu, vloženie dát do textového dokumentu, prípadne zobrazenie údajov v externom textovom editore na celej obrazovke.

### **Ostatné prvky**

Okrem vyššie spomínaných prvkov obsahuje sada nástrojov grafických komponent veľa ďalších užitočných prvkov. Niektoré najčastejšie používané z viditeľných prvkov pre návrh užívateľského rozhrania sú popísané v tejto sekcii.

Ukazovateľ priebehu v podaní tejto sady nástrojov je horizontálny pás, ktorý sa zaplňa v závislosti na sledovanom priebehu. Užívateľ podľa ukazovateľa vie, že aplikácia funguje a môže odhadovať čas priebehu. Implementačne sa jedná o prvok, ktorému sa na začiatku nastaví hodnoty, napríklad maximálna hodnota a počas sledovaného priebehu sa do ukazovateľa posiela aktuálna hodnota prepočítaná na krok ukazovateľa.

Ďalším rozšíreným prvkom sú popisy. Pri návrhu užívateľského rozhrania je možné jednotlivým popisom meniť fonty, veľkosti a iné parametre. Zmenou veľkosti fontu na väčší, prípadne zmenou hrúbky fontu je možné popis použiť ako nadpis.

Pre vizuálne oddelenie jednotlivých funkčných sekcií je možné použiť panely. Panelom je možné priradiť rôzne farby a taktiež tieto farby jednoducho meniť počas chodu aplikácie. Panely teda okrem vizuálneho oddelenia funkčných sekcií dokážu slúžiť ako informačné prvky vyjadrujúce aktuálny stav.

Ostatné viditeľné použité prvky sú napríklad tlačidlá, oddeľovače, zaškrŕavacie políčka, rozbaľovanie listy, zoznamy a ďalšie. Túto sadu nástrojov je v prípade potreby možné rozšíriť pomocou rôznych iných vlastných grafických komponent, vo väčšine prípadov je však možné si vystačiť iba s natívnymi komponentami.

## **3.1.3 Ergonómia používania aplikácie**

Počas vývoja aplikácie neustálou potrebou testovania vzišla najavo hlavná požiadavka pre zvýšenie ergonómie a to ukladanie a načítavanie stavov. Načítanie zadaných stavov šetrí čas. Podobne ako pre vyššie spomínané funkcie na zvýšenie ergonómie, aj táto funkcia pomôže znížiť šancu zavedenia chyby obsluhou aplikácie. Stavby môžu byť počas

chodu načítavané z dátových štruktúr v aplikácií, ale pre možnosť načítavania stavov aj po ukončení aplikácie a jej následnom opätovnom spustení je nutné vytvoriť funkcie pre prácu so súbormi. Funkcia pre uloženie stavov prejde danú dátovú štruktúru a uloží ju do štruktúrovaného súboru. Funkcia pre načítanie súboru býva z pravidla zložitejšia, musí totižto obsahovať algoritmy pre rozbor dát zvyčajne po kúskoch načítavaných zo súboru a ich správne priradenie do dátovej štruktúry.

Pre ukladanie stavov do súborov je vhodné použiť pre každú skupinu stavov inú príponu súborov. Pre načítavanie stavov zo súboru je nutné nastaviť filtrovanie prípon súborov. Užívateľovi sa potom budú ukazovať len relevantné súbory, čo sprehládni výber a zamedzí vybratie nesprávneho typu súboru. Pokiaľ by užívateľ vybral nesprávny typ súboru, mohlo by dôjsť k pádu aplikácie.

Ergonómia používania sa taktiež viaže k jednoduchosti hľadania jednotlivých ovládacích prvkov v užívateľskom rozhraní. K väčšej prehľadnosti je možné použiť vhodné rozdelenie obsahu oddeľovačmi a popísanie sekcií pomocou popisov. Pre rýchlejšiu orientáciu sa medzi jednotlivými sekciami je vhodné samotné sekcie umiestniť do panelov. Panely je možné následne farebne odlíšiť. Použitie farebného rozlíšenia jednotlivých sekcií zabezpečí vyšší kontrast, ako medzi sekciou a pozadím, tak aj medzi samotnými sekciami.

## 3.2 Návrh algoritmov

Vytváranie algoritmov a písanie zdrojového kódu ide paralelne s navrhovaním grafického rozhrania. Pre začiatok je potrebné napísať algoritmy, pre pokrytie základného fungovania jednotlivých prvkov grafického rozhrania, ktoré sa neskôr upravujú a vylepšujú. Optimalizácia a vylepšovanie algoritmov býva zvyčajne proces, ktorý vychádza z potrieb popri skúšaní funkčnosti aplikácie. Používanie aplikácie častokrát vytvára nové požiadavky pre funkcionality algoritmov.

V tejto časti práce sú algoritmy väčšinou popísane vo výslednej podobe vysokoúrovňovým náhľadom. Pre samotnú komplexnosť algoritmov sú častokrát tieto náhľady zjednodušené. Zdrojové kódy majú dokopy niečo málo cez desať tisíc riadkov, z toho dôvodu by nízkoúrovňový detailný popis algoritmov, či už slovný alebo grafický, prevyšoval rozsah tejto práce. Podrobnejším vysvetlením problematiky popisu algoritmov sa zaoberá podkapitola ovládania prvkov grafického užívateľského rozhrania, (z angličtiny *Graphical User Interface* – GUI) 3.2.3 Algoritmy pre ovládanie GUI.

### 3.2.1 Univerzálnosť kódu

Pri písaní objektovo orientovaných zdrojových kódov je vhodné vytvárať univerzálne metódy. Univerzálne pre možnosť opätovného použitia vo viacerých častiach aplikácie. Šetrí to čas pre vývoj a zdrojové kódy sú taktiež prehľadnejšie. Príkladom univerzálnych metód v aplikácií môžu byť metódy na formátovanie a vyfarbovanie tabuliek. Okrem takýchto univerzálnych metód má aplikácia aj ďalšie iné univerzálne metódy, ktoré vznikali pri programovaní aplikácie a často práve kvôli svojej podobnosti boli prepísané ako samostatné metódy.

### 3.2.2 Testovanie a ošetrovanie stavov

Ďalším dôležitým prvkom pri písaní zdrojového kódu je jeho testovanie na rôzne stavy, ktoré môžu nastať. Tieto stavy je nutné ošetriť, zamedzí to chybám vo výslednej aplikácii, prípadne možným pádom aplikácie pri preklepoch v zadávaní dát.

Testovanie stavov prebiehalo pomocou používania rôznych rozličných vstupných údajov. Napríklad testovanie funkčnosti behu aplikácie iba pri zadaní jednej opravnej linky, prípadne pri zadaní maximálneho počtu liniek, pri zadaní jedného úseku na linke, prípadne pri zadaní maximálneho počtu úsekov. Podobne prebiehali aj rôzne iné testovania možných stavov, tieto stavy však boli predpokladateľné a logické.

Aplikácia bola v rámci možností testovaná aj na nepredpokladateľné a nelogické stavy. Takéto stavy môžu vzniknúť nepozornosťou, alebo preklepom obsluhy aplikácie, prípadne rutinou a s ňou spojeným nevšímnutím si neplatnej prednastavenej hodnoty. Príkladom v tomto prípade môže byť zadanie nečíselného údaju do číselného poľa. Pri zadaní nečíselného údaju do číselného poľa grafického rozhrania nastane chyba, ktorá môže zapríčiniť nefunkčnosť celej aplikácie. Tento príklad vygeneruje hlášku o nečakanej neošetrenej výnimke a v závislosti na ďalšom zdrojovom kóde môže poškodiť dátovú štruktúru aplikácie. Väčšinu chýb takéhoto typu je možné odstrániť obalením volania metódy, prípadne väčšej časti zdrojového kódu v takzvanom skús – chyt' (pojem je všeobecne známy pod anglickým názvom *try - catch*) bloku, ktorý odchyť výnimku a na základe jej hodnoty určí správanie aplikácie. Pre tento konkrétny prípad je možné obaliť do skús bloku časť kódu, ktorá prevádza údaj z poľa na číslo a v prípade vrátenia výnimky implementovať do chyt' časti metódu pre vypísanie chybovej hlášky o nesprávnom type údaju v číselnom poli. Chybná hláška upozorní obsluhu na problém a navedie obsluhu na korekciu zdroja problému. Ošetrovanie takéhoto typu chýb býva zväčša jednoduché, avšak pri veľkom množstve ovládacích prvkov môže byť značne časovo náročné.

Hľadanie chýb a testovanie stavov je časovo náročná úloha, ktorá značne zdržuje vývoj aplikácie. Z toho dôvodu pri profesionálnom vývoji aplikácií existujú celé tímy pracovníkov, ktoré sa zaoberajú iba samotným hľadaním a nahlasovaním chýb.

### 3.2.3 Algoritmy pre ovládanie GUI

Potrebné algoritmy pre prácu s grafickým užívateľským rozhraním môžu mať rôzne zložitosti a od toho sa odvíja možnosť ich popisu a reprezentácie. Niektoré je možné detailne popísať krok po kroku na úrovni zdrojového kódu, iné sú veľmi zložité a k ich zrozumiteľnej interpretácii je vhodnejšie použiť vysokoúrovňového a častokrát ešte zjednodušeného popisu. V tejto časti sa preto nachádzajú rôzne typy *Unified Modeling Language* – UML diagramov, vyjadrujúce rôzne úrovne abstrakcie. Pre lepšie pochopenie problematiky popisu algoritmov sú v odstavcoch nižšie uvedené príklady rozličných zložitostí algoritmov obsluhujúcich užívateľské rozhranie.

Užívateľ môže stlačením tlačidla vyvolať napríklad jednoduchú akciu, akou môže byť zmena počtov stĺpcov tabuľky. Algoritmy pre zmenu počtu stĺpcov tabuľky sa využívajú pri vytváraní kostry strediska. Táto zmena obsahuje zmazanie obsahu tabuľky, prefarbenie jednotlivých buniek a zavolanie metódy pre prekreslenie tabuľky. Popis takejto zmeny je možné zobrazit' na relatívne nízkej úrovni, pretože algoritmy pre obsluhu tohto tlačidla nie sú nijak zvlášť rozsiahle.

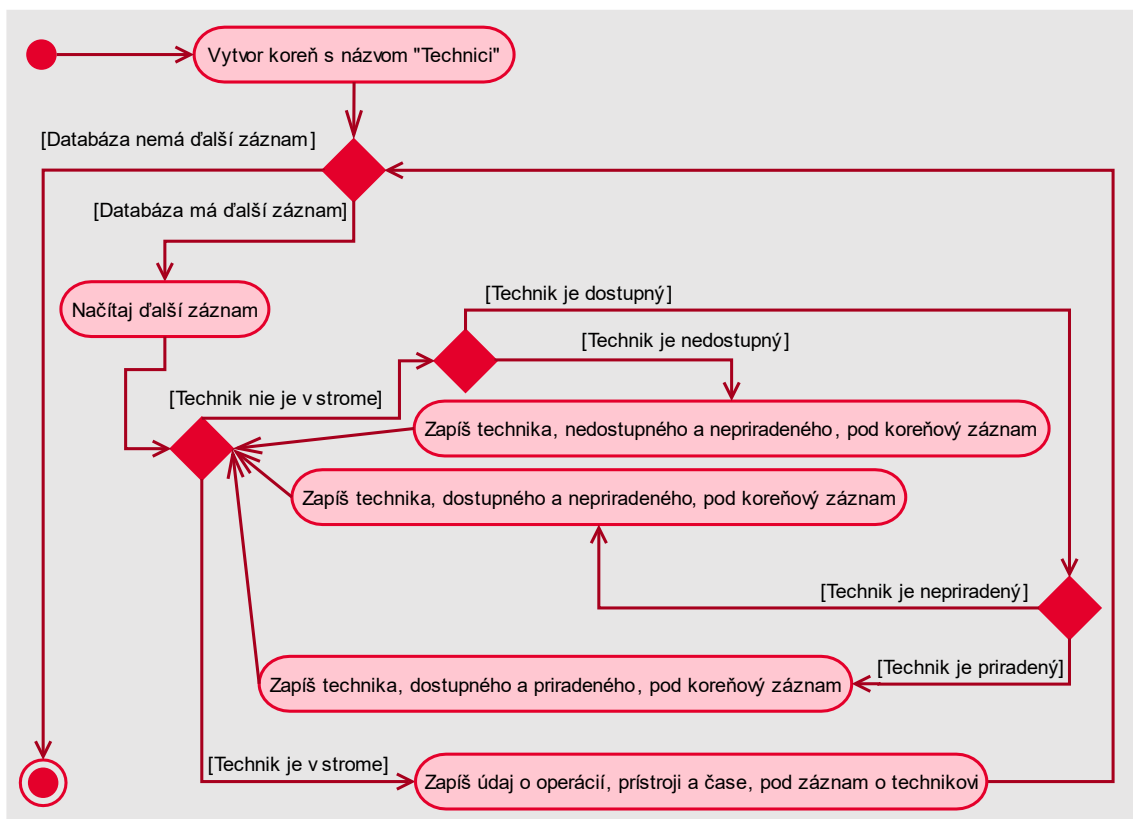
Mierne zložitejšie je napríklad nastavenie parametrov na úseku modelu. Jedným stlačením sa užívateľovi zobrazí ponuka pre vyplnenie rozličných údajov konkrétneho úseku, ktorá si najskôr načíta už uložené parametre, ktoré zobrazí v rôznych prvkov grafického užívateľského rozhrania. Následne môže dôjsť k úprave týchto parametrov, ktorými sú napríklad modelovanie krokov, manipulácia s technikmi, úprava zoznamu prístrojov na úseku, importovanie prístrojov na úseku a podobne. Samotné algoritmy uskutočnia veľa krokov, ktorých popísanie by bolo značne rozsiahle a neprehľadné. Výhodné je preto použiť vyššiu úroveň abstrakcie a nezaoberať sa samotnými krokmi zdrojového kódu.

Situácia je o niekoľko úrovní zložitejšia, pokiaľ užívateľ stlačením tlačidla vykoná napríklad simuláciu. Samotné algoritmy obsluhujúce tlačidlo pre vykonanie simulácie hneď na najvyššej úrovni vykonajú dve samostatné akcie. Skontrolovanie validity zadaných údajov a vykonanie simulácie. Kontrola validity obsahuje na nižšej úrovni ďalšie rôzne algoritmy, ktoré prechádzajú celým strediskom a kontrolujú samotné rozloženie technikov na úsekoch, ich schopnosti pracovať s rôznymi typmi prístrojov a samotné prístroje v stredisku, aby sa nestalo, že simulácia nebude môcť so zadanými údajmi prebehnúť. Taktiež ošetrujú, aby počas simulácie nemohli nastať chyby kvôli neočakávaným stavom. Na najnižšej úrovni samotného zdrojového kódu sa jedná o zložité vnorené cykly, ktoré na základe rôznych podmienok volajú metódy obsahujúce ďalšie zložité vnorené cykly. Samotné simulačné algoritmy sú oveľa náročnejšie a ešte viac rozvetvené, v porovnaní s algoritmami pre kontrolu validitu údajov. Navyše počas behu simulácie taktiež dochádza k upravovaniu grafických prvkov užívateľského rozhrania, konkrétne k dočasnej deaktivácii tlačidla pre spustenie simulácie a taktiež k posúvaniu ukazovateľa priebehu, aby bol užívateľ informovaný o aktuálnom priebehu simulácie.

Dokopy teda algoritmy obsluhujúce tlačidlo pre vykonanie simulácie obsahujú stovky riadkov kódu so značne rozvetvenou rozhodovacou logikou. Z tohto dôvodu nie je možné prehľadne popísať tieto algoritmy nízkou úrovňou abstrakcie, napríklad na úrovni samotného zdrojového kódu.

### **Stromová štruktúra a načítanie údajov o technikoch**

Počas nastavovania parametrov modelu je často potrebné načítať nejaké údaje zo súboru a pre možnosť spätnej kontroly ich nejakou formou vizualizovať. Dáta, ktoré sa načítavajú zo súboru tvoria častokrát rozsiahlu štruktúru a preto sa ako vhodná forma vizualizácie naskytuje práve vykreslenie údajov do stromovej štruktúry. Zobrazenie dát pomocou stromovej štruktúry umožňuje rozvetvovať a skrývať jednotlivé elementy a teda je možné prehľadne prechádzať údaje po vrstvách. Príklad dát vykreslených v stromovej štruktúre je v podkapitole *Prvok stromová štruktúra* (Obrázok 12 a Obrázok 13). Algoritmy obsluhujúce zapisovanie technikov do stromovej štruktúry je možné zobrazit' detailnejšie, síce stále zjednodušene, ale už na úrovni samotného zdrojového kódu (Obrázok 14). Aj napriek výraznému zjednodušeniu diagramu je diagram celkom veľký, aj napriek toho, že sa jedná o triviálny algoritmus. Tento fakt ilustruje nemožnosť použitia nízkoúrovňového popisu pre popis algoritmov aplikácie.



Obrázok 14 Zjednodušený diagram aktivít algoritmu načítania technikov do stromovej štruktúry

Diagram aktivít (Obrázok 14) popisuje zdrojový kód s vyššou abstrakciou, napríklad ako kroky ktoré vykonávajú určitú aktivitu, s dôrazom na rozhodovaciu štruktúru. V tomto diagrame je kladený dôraz na popísanie zdrojového kódu skôr na úrovni funkcionality, ako na úrovni samotných premenných, jeho predchodca je vývojový diagram, ten by však pri popisovaní zložitejších algoritmov nabral rozsiahlych rozmerov. Zápis do stromu na najnižšej úrovni prebieha pomocou metódy stromu *AppendItem* pre pridanie záznamu. Najdôležitejšie parametre tejto metódy sú identifikátor záznamu v strome typu *wxTreeItemId* a textový reťazec, ktorý má byť pod tento identifikátor pridaný. Metóda *AppendItem* taktiež vracia parameter typu *wxTreeItemId* pridaného záznamu. Pokiaľ by mal byť diagram detailnejší, obsahoval by dve premenné typu *wxTreeItemId*, jednu, do ktorej sa uloží návratová hodnota koreňového záznamu a druhú, do ktorej sa ukladá hodnota aktuálneho záznamu technika, získaná ako návratová hodnota pri zapisovaní technika do stromu. Premenná obsahujúca identifikátor koreňa je použitá ako parameter pri pridávaní technikov a premenná, ktorá obsahuje identifikátor technika slúži zas ako parameter pre pridávanie údajov o operácií. Rozhodovacia štruktúra využíva metódy z objektu databázy technikov. Pre kontrolu dostupnosti technika je v triede databázy technikov deklarovaná metóda *jeTechnikDostupny* a pre kontrolu priradenia technika na úseku metóda *jeTechnikPriradeny*. Tvorba textových reťazcov využíva taktiež metódy, ktoré poskytuje databáza technikov. Pre informácie o technikovi, operácií a prístroji sú použité metódy, ktoré vracajú textový reťazec, konkrétne *getTechnik*, *getPristroj* a *getOperacia*. Pre údaj o čase slúži metóda *getDoba*, ktorá však vracia údaj v číselnom formáte. Prevedenie číselného údaju na textový je v algoritme využitá metóda *to\_string* zo štandardnej knižnice.

Vizualizácia dát do stromovej štruktúry je v aplikácii taktiež využitá pri nastavovaní stavov modelu. Pri práci so stavmi modelu slúži pre náhľad na aktuálne stavy a parametre nastavené pomocou mapy modelu. Načítavanie údajov do stromovej štruktúry pre tento prípad prebieha pri nejakej zmene stavu, alebo parametra modelu, či už manuálnej, alebo automatizovanej, vyvolanej priradeniami a optimalizačnými algoritmami. Tento strom je popísaný nasledujúcimi odstavcami. Pre lepšie pochopenie je doporučené si najskôr pozrieť ukážku kúsku tohto stromu zobrazenú v podkapitole *Prvok stromová štruktúra* na obrázku využitia stromovej štruktúry (Obrázok 12).

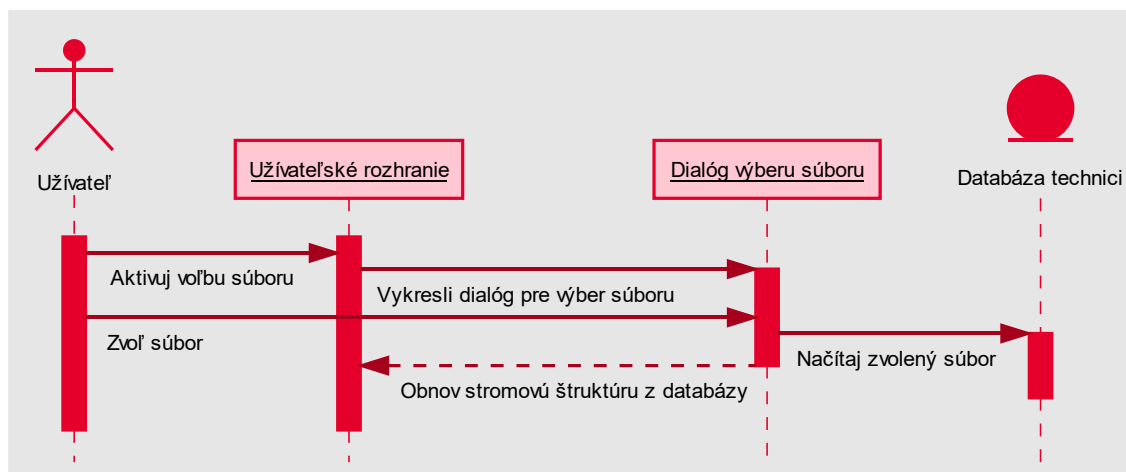
Strom vizualizácie stavov strediska je tvorený koreňom s informáciou o počte liniek v stredisku. Z koreňa ďalšiu úroveň tvoria samotné linky, od prípravnej linky, až po dokončovaciu linku. Ďalšia úroveň stromu tvorí zoznam úsekov na linkách. Pre každý úsek existuje nižšia úroveň zložená z troch vetiev – model operácie na úseku, počet priradených technikov a počet čakajúcich prístrojov na úseku. Tieto vetvy môžu nadobudnúť dva stavy. Ak obsahujú dodatočné informácie, tak sa ďalej rozvetvia, ak nie, tak sa nimi stromová štruktúra ukončuje.

Vetva modelu operácie na úseku môže byť jedna z koncových vetiev. Pokiaľ úsek spočíva v jednej celistvej operácii vykonanej technikom, táto vetva je koncová s informáciou, že operácia na úseku je jeden súvislý blok. Pokiaľ však model úseku obsahuje viacero krokov, ktoré technik musí vykonať, vetva sa ďalej rozvetvuje na koncové vetvy, ktoré v názve obsahujú informácie o samotných krokoch. Informácie obsiahnuté pri názve kroku sú – koľko percent času technika vykonávajúceho operáciu zaberie konkrétny krok, prípadne ak je krok automatizovaný, informáciu, že sa jedná o krok časovo nezávislý na technikovi.

Vetva s technikmi priradenými na operáciu môže byť koncová s informáciou o nepridelenom, prípadne jednom pridelenom technikovi, alebo sa rozvetví na koncové vetvy so zoznamom technikov. Podobne je na tom aj vetva so zoznamom prístrojov. Pokiaľ na úseku nečakajú prístroje je vetva koncová, pokiaľ však na úseku prístroje čakajú, vetva sa ďalej rozvetví na koncové vetvy so zoznamom prístrojov.

Pri porovnaní výslednej štruktúry dát toho stromu so štruktúrou dát stromu s údajmi o technikoch popísaných pod obrázkom so zjednodušeným diagramom aktivít načítania technikov do stromovej štruktúry (Obrázok 14) je možné usúdiť, že algoritmy obsluhujúce zápis údajov do stromu stavov strediska sú násobne komplikovanejšie. Okrem viacnásobného počtu rozhodovacích štruktúr musia obsahovať pre jednotlivé úrovne rovnako aj viacnásobný počet premenných typu *wxTreeItemId*. Vizualizácia týchto algoritmov by teda vytvorila na rovnakej úrovni abstrakcie niekoľkonásobne väčší neprehľadný diagram.

Príklad návrhu celého postupu pre načítanie údajov o technikoch zo súboru do stromovej štruktúry je uvedený na diagrame nižšie (Obrázok 15). Vyššie spomínané algoritmy pre zápis do stromovej štruktúry sú v tomto sekvenčnom diagrame skryté vyššou úrovňou abstrakcie. Konkrétne algoritmy pre načítanie technikov do stromovej štruktúry sú použité pri načítaní súboru s údajmi o technikoch po ukončení dialógu pre výber súboru. Sú volané vrátením správy pre obnovenie stromovej štruktúry z databázy technikov, ktorú posiela objekt dialógu výberu súborov.



Obrázok 15 Sekvenčný diagram načítania technikov od výberu súboru, po načítanie do stromovej štruktúry, zjednodušená verzia

Vyššie uvedený sekvenčný diagram (Obrázok 15) popisuje interakciu užívateľa s aplikáciou a jednotlivé kroky pre načítanie technikov spolu s ich údajmi zo súboru. Sekvenčný diagram je taktiež jeden z typov UML diagramov pre popis objektovo orientovaného zdrojového kódu. Reprezentuje činnosti v závislostiach na časovej postupnosti. Diagram na nižšej abstraktnej úrovni môže znázorňovať jednotlivé volania medzi objektami, na úrovni s vyššou abstrakciou zas jednotlivé činnosti. Čítanie diagramu je zľava doprava a zhora smerom dole.

V prvom kroku užívateľ aktivuje voľbu súboru stlačením tlačidla pre výber súboru v časti užívateľského rozhrania s údajmi o technikoch. Grafické rozhranie vykreslí dialógové okno pre výber súboru. Užívateľ vyberie súbor, cesta vybraného súboru sa predá entite databázy technikov ako parameter metódy *loadFileTechnici*, ktorá načíta technikov a ich údaje zo súboru do objektu databázy. Algoritmy pre načítanie technikov fungujú na princípe rozdelenia štruktúry súboru na jednotlivé údaje, ktoré sú priradované do dátovej štruktúry entity databázy. Zavretím okna užívateľské rozhranie dostane pokyn k prekresleniu stromovej štruktúry technikov. Toto prekreslenie sa následne vykoná pomocou algoritmov popisovaných odstavcami vyššie a taktiež v grafickej forme diagramom na obrázku zjednodušeného diagramu aktivít načítania technikov do stromovej štruktúry (Obrázok 14).

### Nastavenie úseku na mape modelu strediska

Pre čo najbližšie priblíženie priebehu a s ním spojeného výsledku simulácie reálnemu chodu strediska je nutné vytvoriť čo najpresnejší model. Aplikácia v tomto kroku už teda má vytvorenú kostru strediska, v ktorej sú jednotlivé linky a samotné úseky na linkách, podrobnejšie kostrou strediska sa práca zaoberá v podkapitole *Modelovanie kostry strediska*. V reálnom stredisku sa však niektoré úseky neskladajú len z jednej technikom vykonávanej operácie, ale táto operácia býva častokrát prerušovaná rôznymi automatizovanými procesmi, ktoré na technikovi nie sú závislé a teda technik počas čakania na vykonanie prechodu automatizovaným procesom môže začať prácu na ďalšom prístroji. Prvou požiadavkou na nastavenie úseku je teda možnosť namodelovania jednotlivých krokov. V aplikácii sú automatizované procesy nazvané ako medzi – operácie a je možné nimi rozdeľovať celkový čas technika na operácií.

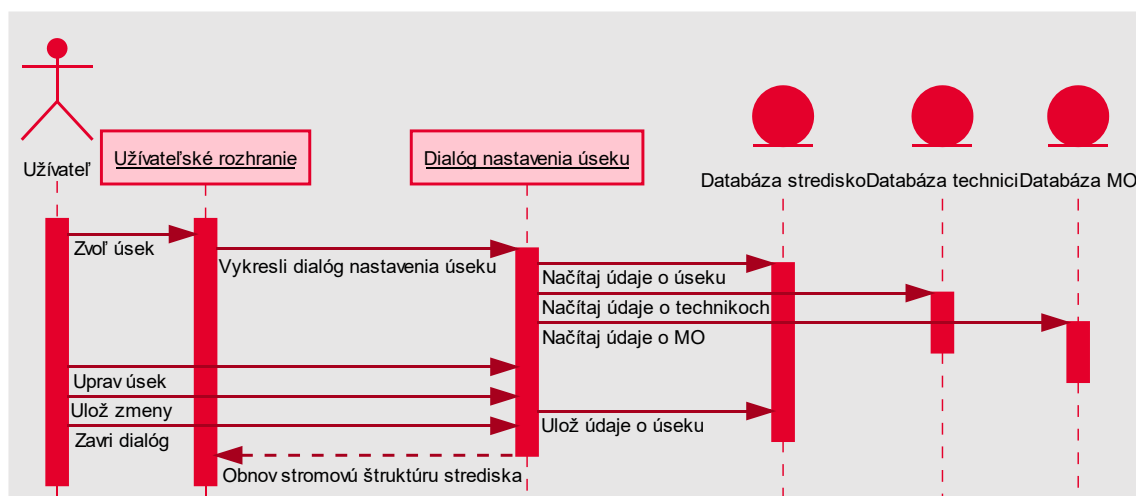
Ďalšou požiadavkou pre nastavenie úseku je samozrejme možnosť ručne priradovať technika na úsek a možnosť nastavenia limitácie maximálneho počtu technikov na úseku. Nastavenie limitácie počtu technikov je využívané pri automatizovaných priradovacích a optimalizačných algoritmov, aby nedošlo k priradeniu väčšieho počtu technikov, ako je počet pracovísk na úseku. Pre možnosť manuálneho zásahu do optimalizovaných dát je potrebné taktiež pridať možnosť zmazať zvoleného technika z úseku.

Poslednou požiadavkou pre nastavenie úseku je možnosť pridania prístrojov s číslami zákaziek. Pre komfortnejšie ovládanie aplikácie je vhodné prisať možnosť načítania zoznamu zákaziek na úsek zo súboru. Užívateľ potom nemusí manuálne pridávať prístroje a zapisovať im priradené čísla zákaziek. V prípade chybného zadania zákazky je dobré pridať možnosť editovať číslo zákazky, prípadne zmazať konkrétny prístroj. Pridanie prístrojov na prvý úsek prvej linky značí prístroje čakajúce na vstupe strediska. Prístroje na ostatných úsekoch, ktoré sa v stredisku nachádzajú pred spustením simulácie sú rozpracované prístroje (označované taktiež skratkou WIP). Zhrnutie týchto požiadaviek je uvedené v tabuľke nižšie (Tabuľka 6).

Tabuľka 6 Zhrnutie požiadaviek pre nastavenie úseku

Nastavenie technikov	Nastavenie prístrojov	Nastavenie operácie
Pridanie technika	Pridanie prístroja	Limitácia počtu technikov
Zmazanie technika	Vyplnenie čísla zákazky	Rozdelenie na kroky
Zmazanie celého zoznamu	Editovanie čísla zákazky	Pridanie MO medzi kroky
	Zmazanie prístroja	Zmazanie modelu operácie
	Zmazanie celého zoznamu	
	Import zoznamu zo súboru	

Pre vizualizáciu nastavenia úseku je opäť možné využiť sekvenčný diagram. Vytvorený diagram je na obrázku nižšie (Obrázok 16). V porovnaní so sekvenčným diagramom načítania technikov zo súboru (Obrázok 15) je tento diagram zložitejší a taktiež viac zjednodušený.



Obrázok 16 Sekvenčný diagram interakcie pre nastavenie úseku, zjednodušená verzia



Nastavenie parametrov úseku sa začína výberom úseku na mape tvorenej prvkom tabuľka v hlavnom okne aplikácie. Príklad takejto mapy je uvedený v podkapitole *Prvok tabuľka* na obrázku Ukážka kúsku tabuľky so skrytými popismi využitej na vykreslenie mapy modelu (Obrázok 9). Užívateľ stlačením príslušnej bunky v tabuľke vyberie úsek. Následne dôjde k vykresleniu dialógu nastavenia úseku, ktorý si načíta hodnoty z databáz. V diagrame (Obrázok 16) sú uvedené tri databázy, s ktorými dialógové okno pracuje. Databáza strediska, technikov a medzi – operácií.

Databáza strediska obsahuje celú štruktúru strediska, linky, úseky, spolu s aktuálnymi stavmi úsekov. Táto databáza sa využíva na načítavanie údajov o konkrétnom úseku, aby bolo možné automaticky vyplniť dialóg nastavenia úseku aktuálnymi hodnotami a užívateľ nemusel nastavovať celý úsek nanovo. Databáza strediska sa taktiež používa pri ukladaní, ktoré užívateľ zahájí stlačením tlačidla uložiť. Ukladacie algoritmy prechádzajú grafickým rozhraním dialógového okna a pomocou dielčích algoritmov ukládajú hodnoty z grafických prvkov do štruktúry databázy strediska.

Ďalšie dve databázy slúžia len na počiatočné načítanie údajov a v tejto časti sa do nich nezapisuje. Databáza s technikmi slúži na načítanie menného zoznamu technikov a taktiež pre získanie zoznamu prístrojov. Databáza s automatizovanými procesmi slúži pre získanie zoznamu názvov týchto procesov. Tieto zoznamy sa používajú pre doplnenie prvkov užívateľského rozhrania v dialógovom okne prednastavenými hodnotami. Užívateľ si tak môže vyberať technikov, prístroje a medzi – operácie zo zoznamov a nemusí ručne písať celé názvy.

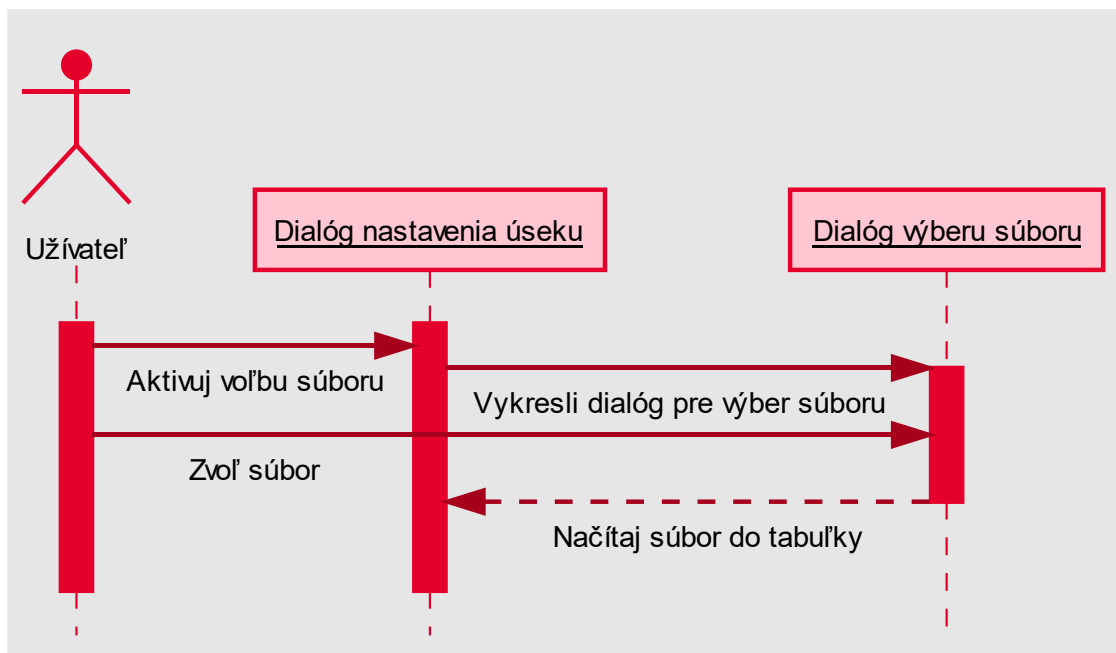
Dialógové okno obsahuje rôzne prvky pre tvorbu užívateľského rozhrania, od najčastejšie používaných prvkov tabuľky, stromovej štruktúry, tlačidiel, popisov, textových polí obsahuje aj zaškrŕavacie poličko, rolovací zoznam pre výber možností a kombinované pole.

Tabuľka je použitá pri modelovaní krokov operácie na úseku a taktiež pri zapisovaní prístrojov spolu so zákazkami do zoznamu prístrojov na úseku. Týmto je možné dosiahnuť jednu z požiadaviek a to konkrétne editáciu čísla zákazky prístroja. Názov prístroja je napísaný v tabuľke ako needitovateľný názov riadku a číslo zákazky v editovateľnej bunke príslušného riadku.

Stromová štruktúra je použitá len jednoúrovňová, so skrytým koreňom, ako zoznam prístrojov načítavaný z databázy technikov. Tieto prístroje je možné stlačením tlačidla pridávať do tabuľky zoznamu prístrojov na úseku. Výhodou oproti použitiu kombinovaného poľa, alebo rolovacieho zoznamu je v prehľadnosti a rýchlosti výberu. Nevýhodou je na druhej strane nemožnosť vyhľadávania pri veľkých zoznamoch a taktiež pre veľké zoznamy veľá vertikálneho miesta. Taktiež je nevhodná na zobrazovanie dlhých názvov, pretože by zaberala veľkú plochu. Prístroje nemajú dlhé názvy a taktiež netvorí dlhý zoznam. Pre výpis zoznamu prístrojov je použitie stromovej štruktúry vhodná voľba.

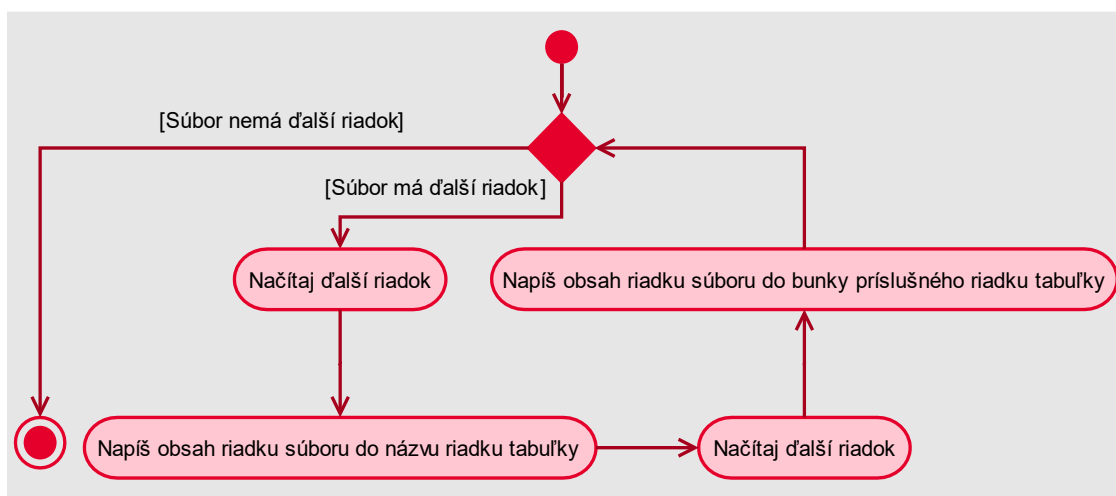
Rolovací zoznam je využitý na voľbu automatizovaných procesov. Výhodou je, že nezaberie veľá miesta a oproti kombinovanému poľu je prehľadný. Rovnako ako kombinované pole zaberie miesto jedného riadku a dokáže zobrazíť dlhé názvy. Nevýhodou oproti kombinovanému poľu je nemožnosť vyhľadávania. V jednotlivých úsekoch sa však zvyčajne nenachádza veľá automatizovaných procesov a preto je jeho použitie pre tento účel výhodnejšie.

Kombinované pole je využité pre výpis zoznamu technikov. Aj napriek tomu, že sú technici filtrovaní a zobrazujú sa iba technici, ktorí vedú na danom úseku pracovať (Obrázok 19), vytvárajú pomerne dlhý zoznam. V kombinovanom poli je možné meno technika napísať a taktiež daného technika jednoducho vyhľadať.



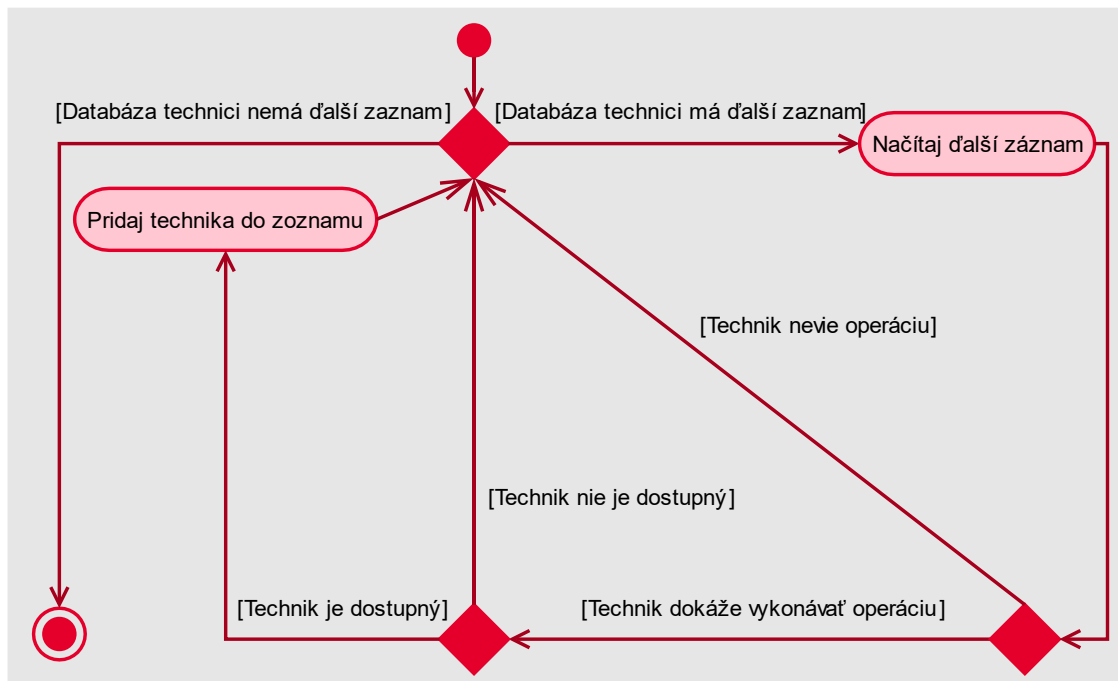
Obrázok 17 Zjednodušený sekvenčný diagram načítania prístrojov zo súboru do tabuľky

Vyššie uvedený diagram vyjadruje interakciu užívateľa s aplikáciou pri importovaní zoznamu prístrojov a čísel zákaziek do tabuľky (Obrázok 17). Diagram je na tejto úrovni abstrakcie podobný diagramu načítania technikov zo súboru do stromovej štruktúry (Obrázok 15), akurát bez ukladania údajov do databázy, s priamym načítaním údajov zo súboru do tabuľky. Na nižšej úrovni už však bude vidieť viac rozdielov. Na nižšej úrovni rozpísané algoritmy pre načítanie súboru sú na diagrame nižšie (Obrázok 18) a v porovnaní s diagramom aktivít pre načítanie technikov do stromovej štruktúry (Obrázok 14) sú výrazne jednoduchšie.



Obrázok 18 Diagram aktivít pre načítanie zoznamu prístrojov zo súboru

Aplikačný diagramu načítania zoznamu prístrojov so zákazkami zo súboru do tabuľky (Obrázok 18) je v aplikácii najjednoduchší algoritmus z algoritmov pre spracovanie štruktúry súboru. Súbor obsahuje po riadkoch rozdelené údaje, postupne riadok po riadku, najskôr názov prístroja a nasledujúci riadok je číslo zákazky.



Obrázok 19 Zjednodušený diagram aktivít pre pridávanie dostupných spôsobilých technikov

Diagram aktivít uvedený na obrázku vyššie (Obrázok 19) popisuje filtrovanie technikov pri načítavaní do menšieho zoznamu v kombinovanom poli.

Testovanie technika pre spôsobilosť vykonávania operácie na úseku zabezpečuje v podmienke volanie metódy vracajúcej binárnu hodnotu. Parametrami metódy sú textové reťazce s menom technika a názvom operácie. Metóda je definovaná v triede databázy technikov, prechádza jednotlivé záznamy vo vektore technikov a pokiaľ nájde záznam, ktorý určuje, že technik dokáže vykonávať túto operáciu vráti hodnotu *true*. Pokiaľ metóda nenájde záznam o schopnosti technika danú operáciu vykonávať, vráti hodnotu *false*.

Dostupnosť technika je binárna premenná definovaná vo vektore technikov v triede databázy technikov. Pre použitie v podmienke je v databáze technikov definovaná metóda, ktorá vo vektore nájde technika a vráti hodnotu tejto premennej. Parametrom metódy je textový reťazec s menom technika. Tento parameter má prednastavenú hodnotu *true* a dá sa jednoducho manuálne zmeniť v časti aplikácie s nastavením stavov modelu podľa aktuálnej dostupnosti technikov.

### 3.2.4 Kvantifikovanie liniek a úsekov

Pre možnosť plánovania chodu strediska je pre začiatok nutné uskutočniť nejakú kvantifikáciu, číselné vyjadrenie aktuálneho modelu. Bez číselného vyjadrenia nie je možné predvídať chod strediska, posudzovať efektivitu rozloženia technikov a vyhľadávať slabé miesta v rozmiestnení vhodné na optimalizáciu.

Počas návrhu aplikácie bolo zvažovaných viacero konceptov, z ich skúšania však vyplynulo, že najúčinnější kvantifikácia je pre odhad chodu strediska výpočet intervalov úsekov, liniek a intervalu celého strediska. Pre možnosť posudzovania časových náročností analyticky aplikáciou je vhodné využívať váhy úsekov a liniek.

Pre optimalizačné algoritmy, ktoré pracujú s premiestňovaním technikov sa ukázal ako vhodný kvantifikátor koeficient odobratia a pridania technika. Pomocou tohto koeficientu je možné posúdiť, aký veľký dopad na úsek bude mať odobratie, alebo pridanie technika. Odobratie technika totižto v niektorých prípadoch pri malom počte technikov na úseku môže aj viac ako dvojnásobne (v závislosti na výkonnosti technika) znížiť výkonnosť úseku a tým výrazne rozhodiť celý model. V opačnom prípade, pokiaľ je na úseku veľa technikov, odobratie technika nemusí priniesť výrazne nepriaznivú zmenu výkonnosti na tomto úseku a jeho presunutie na iný úsek môže výrazne zrýchliť priebeh zákaziek strediskom.

Všetky tieto vzorce musia brať do úvahy dynamický charakter servisného strediska a úzko špecifické požiadavky s ním spojené. Príkladom môžu byť rozdielne fronty prístrojov na vstupe. Pokiaľ by bola fronta zložená z prístrojov, ktorých operácie budú značne časovo náročné, budú tieto intervaly iné, ako keby bola fronta na vstupe zložená z prístrojov s krátkymi dĺžkami operácií.

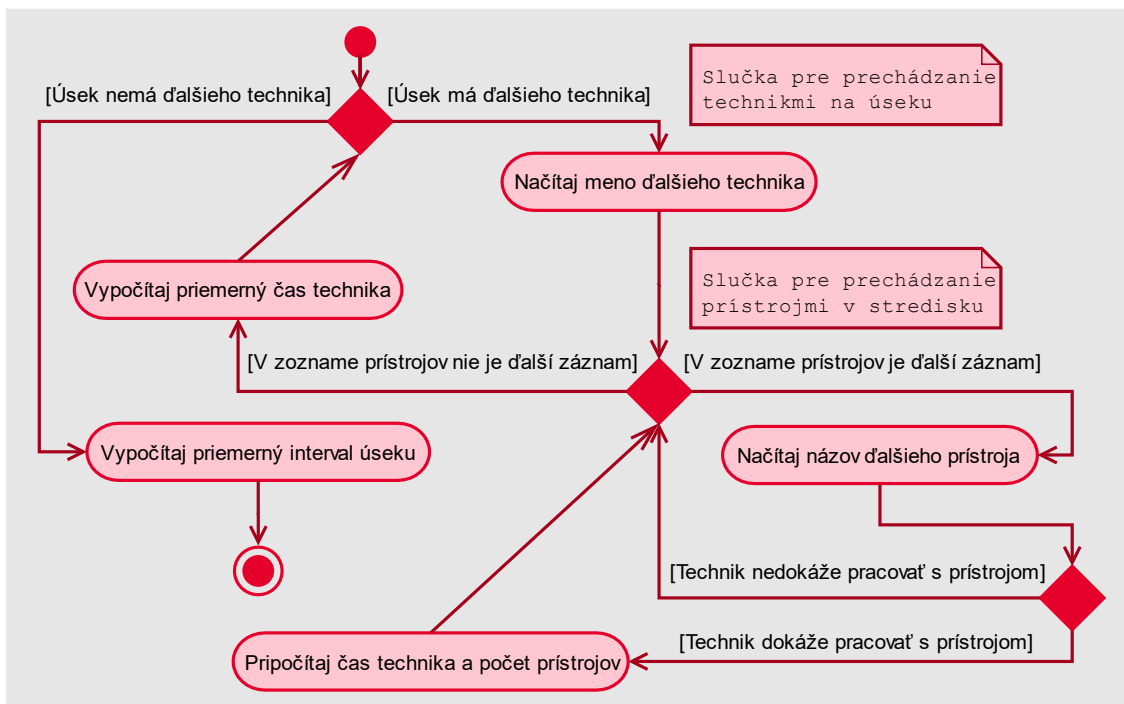
Situácia sa začne komplikovať, keď sa na vstupe začnú objavovať prístroje, ktorých časová náročnosť pre operácie nemá rovnakú distribúciu, ako distribúcia pre ostatné prístroje. Napríklad prístroj, ktorý má všetky dĺžky operácií rovnako dlhé, ale operáciu rozobratie má jednoduchšiu a tým pádom kratšiu. Pokiaľ by takýchto prístrojov bola polovica vstupnej fronty, úsek s operáciou rozobratie by nebol dobre nastavený a dochádzalo by na ňom k menšiemu vyťaženiu technikov. Ak by na druhej strane existoval prístroj, ktorý by mal všetky operácie rovnako dlhé ako ostatné prístroje a operáciu rozobratie by mal dlhšiu, ako ostatné prístroje, úsek s operáciou rozobratie by bol preťažený, dochádzalo by na ňom k hromadeniu a do ďalších úsekov strediska by sa nedostávalo dostatočného počtu prístrojov.

Príklad pre rozdielnu distribúciu rozdielov časov rovnakých operácií na rôznych prístrojoch ďalej komplikuje pseudonáhodné rozdelenie typov prístrojov vo vstupnej fronte. Toto je typický príklad špecifickej vlastnosti servisného strediska. Prístroje vo vstupnej fronte nemajú žiadnu systematickú postupnosť, vstupné fronty fungujú podľa princípu FIFO, primárne tak, ako sú dostávané požiadavky od zákazníkov. Fronty nie je možné ani následne zoradiť pre vytvorenie viacerých optimálnych scenárov pre konkrétne typy prístrojov, pretože to by malo negatívny dopad na čakanie zákazníkov.

Z tohto vyplýva, že výpočty nezáležia len na parametroch rozloženia technikov a ich rýchlostiach pre jednotlivé operácie, ale taktiež musia byť závislé aj na aktuálnych prístrojoch, ktoré vo frontách čakajú na spracovanie. Pre čo najpresnejší výsledok odhadov nie je možné zanedbať akékoľvek špecifické správanie a vytvoriť len zjednodušené algoritmy.

### Algoritmy pre výpočet intervalov

Pre posudzovanie výkonnosti a odhad prechodu bolo treba stanoviť určitý takt, ktorý bude popisovať priemernú dobu prechodu prístrojov. Táto doba je špecifická na typoch prístrojov vo fronte a teda hlavná požiadavka musí byť možnosť prepočtu intervalov v závislosti na zmene prístrojov v stredisku. Interval úseku je závislý na výkonnosti technikov. Dĺžky automatizovaných procesov navýšia čas prechodu úsekom, neovplyvňujú však interval úseku.



Obrázok 20 Diagram aktivít výpočtu intervalu úseku, zjednodušená verzia

Zjednodušený diagram na obrázku vyššie (Obrázok 20) popisuje výpočet intervalu na úseku. Tento výpočet aplikácia používa na výpočet všetkých úsekov a od intervalov úsekov sa odvíjajú intervaly liniek a aj interval samotného strediska. Diagram obsahuje dve slučky, jednu, ktorá prechádza technikmi na úseku a druhá, vnorená, ktorá následne prechádza zoznam prístrojov v stredisku.

Prechádzaním zoznamu prístrojov sa spočítavajú časy podľa aktuálnych počtov prístrojov v stredisku. Každou iteráciou sa testuje schopnosť konkrétneho technika pracovať s daným prístrojom, aby nedošlo k skresleniu výpočtov. Priemerný čas technika je aritmetický priemer časov technika pre prístroje v zozname s ktorými dokáže pracovať. Z týchto časov sa taktiež aritmetickým priemerom vypočíta priemerný čas technikov a z neho sa následne počíta priemerný interval úseku. Priemerný interval vyjadruje podiel priemerného času technikov a počtu technikov na úseku.

Priemerný interval vlastne vyjadruje ako často z úseku vyjde jeden spracovaný prístroj a teda ako často by do linky mal jeden prístroj pribudnúť. Toto tvrdenie je samozrejme platné iba pri saturovanej linke, kedy pracujú všetci technici. Saturácia liniek je znázornená v podkapitole *Analýza prechodu prístrojov*.

### **Algoritmy pre výpočet váh**

Váha úseku vyjadruje, ako moc je interval úseku v porovnaní s intervalmi ostatných úsekov rozdielny. Pomocou váh úsekov a liniek je možné kvantifikovať prioritu. Zoradením váhy od najväčšej po najmenšiu aplikácia získava poradie váh, ktoré využívajú optimalizačné algoritmy. Pomocou hodnoty váh je taktiež možné posúdiť efektivitu rozloženia technikov. Nasledujúci odstavec popisuje výpočty váh úsekov. Podobné postupy je však možné aplikovať na výpočet váh liniek, pojmy úsek a linka sú v nasledujúcom odstavci zameniteľné.

Výpočet váhy úseku je pomerne jednoduchý. Algoritmus najskôr nájde úsek s najvyšším intervalom, hodnotu tohto intervalu si uloží do premennej. Samotný výpočet je podiel aktuálnej hodnoty intervalu úseku s hodnotou najdlhšieho intervalu z premennej. Úsek s najdlhším intervalom má teda váhu jedna a ostatné úseky s kratšími intervalmi majú váhy menšie ako jedna. V ideálnom prípade by mali mať všetky úseky váhu jedna, to by znamenalo, že všetky úseky majú rovnako veľké intervaly.

Zo samotných hodnôt váh je možné predpokladať hromadenie, prestoje a sledovať nerovnomernosť rozloženia pracovnej sily. Pokiaľ je hodnota nasledujúcej váhy vyššia, na úseku, alebo linke bude dochádzať k hromadeniu, vyťaženosť technikov však bude maximálna. Ak však ďalšia váha bude mať nižšiu hodnotu, ďalšia časť spracováva prístroje rýchlejšie a vznikajú na nej prestoje – prázdne miesta, kedy technici čakajú na príchod prístrojov na vstup.

Pokiaľ by bola hodnota ďalšej váhy rovnaká, v ideálnom prípade by boli technici maximálne vyťažení a zároveň by na úseku nedochádzalo ani k hromadeniu. V reálnom prípade však k istému hromadeniu, alebo prestojom aj napriek rovnakej hodnote váhy dochádzať bude. Dôvodom je závislosť na fronte prístrojov a taktiež na schopnostiach technikov pracovať s prístrojmi. Napríklad pokiaľ technik pracujúci na úseku skončí prácu na prístroji, pošle prístroj ďalej na ďalší úsek a do fronty na vstup tohto úseku príde prístroj, s ktorým technik nevie pracovať bude mať tento technik aj vzhľadom na rovnakú váhu predchádzajúceho úseku prestoj minimálne o veľkosti jedného intervalu predchádzajúceho úseku. Pokiaľ však ani za interval nepríde prístroj, s ktorým technik vie pracovať čakanie by sa predĺžilo o ďalší interval. V extrémnom prípade pri monotónnej fronte zloženej len z prístrojov, s ktorými technik nevie pracovať, technik prácu na úseku ani nezačne. Toto je však extrémny prípad, ktorý v reálnom chode zvyčajne nenastane.

### **Výpočet koeficientov odobratia a pridania technika**

V predchádzajúcej kapitole bol popisovaný kvantifikátor popisujúci rozdielnosť intervalov úsekov. Určuje váhu úseku v porovnaní s ostatnými úsekmi. Pre rozhodovaciu štruktúru pri optimalizovaní bolo potrebné nájsť ďalší ukazovateľ. Ukazovateľ, ktorý by kvantifikoval nie len váhu úseku, ale zároveň aj akúsi váhu zmeny počtu technikov. Ukazovateľ, ktorý dokáže vyjadriť, aká veľká zmena intervalu by mohla nastať, pokiaľ by algoritmy odstránili technika z úseku, prípadne pokiaľ by algoritmy na tento úsek technika pridali.

Pomocou takýchto koeficientov je možné smerovať ďalšie kroky optimalizácie. Nemalo byť totižto zmysel odstrániť z nejakého úseku technika a za účelom zníženia intervalu iného úseku ho dosadiť na tento iný úsek, pokiaľ by interval predchádzajúceho úseku po odobratí technika dokonca prevyšoval pôvodný interval tohto upravovaného, pomalšieho úseku.

Výpočet je orientačný, zato taktiež závislý na fronte zariadení v stredisku, počítaný podobne, ako interval, z priemerných hodnôt časov technikov. Priemerné hodnoty časov technikov sú podrobnejšie rozobraté v podkapitole *Algoritmy pre výpočet intervalov* a na obrázku (Obrázok 20). Samotný výpočet prebieha pomocou dvoch jednoduchých vzorcov, ktoré na základe súčasných dát predpokladajú, ako by orientačne mohol byť interval zmenený pridaním, prípadne odobratím technika. Počet technikov je počet technikov na danom úseku, pre ktorý sa koeficienty počítajú. Použité vzorce sú uvedené nižšie:

$$Koeficient_{odobratia} = \frac{Priemerný\ čas\ technikov}{Počet\ technikov-1} - \frac{Priemerný\ čas\ technikov}{Počet\ technikov} \quad (1)$$

$$Koeficient_{pridania} = \frac{Priemerný\ čas\ technikov}{Počet\ technikov} - \frac{Priemerný\ čas\ technikov}{Počet\ technikov+1} \quad (2)$$

Vzorce pracujú na princípe vypočítania intervalov úseku s aktuálnym počtom technikov a na odhade intervalu, za predpokladu, odobratia, alebo pridania technika. Odchýlka tohto odhadu spočíva v tom, že pridaný technik, prípadne odobratý technik môže mať svoj priemerný čas iný, ako ktorý predpokladá vzorec, čím dôjde k zmene celkovej hodnoty priemerného času technikov na úseku. Táto odchýlka je v porovnaní k samotným rozdielom intervalov úsekov malá, zvyčajne aj o dva rády nižšie a teda odchýlku je možné pri výpočtoch zanedbať.

### Korekcia konštánt

Váhy, intervaly a koeficienty sú veľmi dobrým ukazovateľom. Pri modelovaní strediska je však ako voliteľný parameter možnosť zadať štatistické percentuálne údaje o počte kusov prechádzajúcich z linky pre prípravu do opravných liniek. Viac o tomto parametre je spomenuté v podkapitole *Modelovanie kostry strediska*. Z toho dôvodu je potrebné vykonať určitú korekciu váh liniek.

Ako príklad je možné uviesť situáciu, keď by niektorá linka v oprave mala najdlhší interval a tým pádom najvyššiu váhu, lenže ak ňou štatisticky prejde len pár percent prístrojov, nebude tak vytážená, ako ostatné linky. Pridávanie technikov by skrátilo vypočítaný interval a znížilo váhu linky a úsekov, lenže pridaní technici by nikdy nemuseli byť vytážení.

Počas návrhu korekcie boli vyskúšané rôzne metódy a postupy, najvhodnejším riešením tejto korekcie sa však ukázalo byť vynásobenie týchto konštánt prevedením percentuálneho údaju na pravdepodobnosť v intervale  $\langle 0,1 \rangle$ . Pri skúšaní funkčnosti korekcie boli testované prípady prechodu pre rôzne percentuálne údaje, aby bolo možné posúdiť relevantnosť tejto korekcie. Výpočty a optimalizačné algoritmy s touto korekciou pracovali správne pre celý percentuálny rozsah.

### 3.2.5 Odhad času simulácie

Tento odhad času pracuje na statických výpočtoch prechodu fronty prístrojov skúmaným systémom. Jeho hlavnou výhodou je rýchlosť. Samotný odhad trvá len zlomok sekundy, zatiaľ čo reálny čas spočítaný dynamickou simuláciou trvá niekoľko desiatok sekúnd. Práve pre svoju rýchlosť sa v aplikácii využíva pri odhade doby prechodu počas nastavovania stavov modelu, kde po každej zmene prepočíta nový odhad. Užívateľovi slúži ako ukazovateľ, podľa ktorého môže odhadnúť a nastaviť potrebnú dĺžku dynamickej simulácie.

Veľká rýchlosť algoritmu je však na úkor presnosti. Do výpočtov nie je započítaná žiadna dynamika chodu systému a algoritmy k výpočtu využívajú iba koeficienty a konštanty popísané v predchádzajúcej kapitole. Vo väčšine prípadov je však presnosť výpočtu veľmi dobrá a odchýlka od reálnej dynamickej simulácie býva len v rozmedzí pár minút. Menej častou, ale stále dosť vyskytujúcou sa odchýlkou je odchýlka v rozmedzí pár hodín. Takáto presnosť je pre odhad taktiež postačujúca. Obvykle užívateľ volí dĺžku simulácie ako dĺžku tohto odhadu, ku ktorej ešte pripočíta pár hodín smerom hore, aby sa uistil, že simuláciou prejdú všetky prístroje.

V extrémnych prípadoch zvyčajne môžu nastať dva typy odchýlky. Tieto odchýlky nie sú moc časté, vyskytujú sa zvyčajne pri výrazne nevyváženom rozložení technikov. Takéto rozloženie môže nastať pri manuálnom odobratí technikov, v prípade, keď užívateľ následne nepoužije optimalizačné algoritmy, alebo keď užívateľ zámerne nastaví rozloženie technikov, ktoré výrazne rozhodí veľkosti a nadväznosti intervalov (nadväznosť intervalov rozoberá podrobnejšie podkapitola 4.1.4 Simulácia chodu strediska). Dynamická simulácia chodu strediska však všetky tieto odchýlky spoľahlivo odhalí. Užívateľ následne môže pomocou vizualizácií odhaliť problémový úsek a manuálne vykonať nápravu, prípadne sa môže rozhodnúť pre pohodlnejšiu variantu oproti manuálnym zmenám a pomocou stlačenia jedného tlačidla môže spustiť automatickú optimalizáciu, ktorá už problém nájde a automaticky napravi.

Prvý typ extrémnej odchýlky, ktorý je bežnejší zapríčiňuje odchýlku o veľkosti pár desiatok hodín. Táto odchýlka býva zvyčajne kladná a teda odhad ukazuje o pár desiatok hodín vyšší čas, ako aký v skutočnosti bude. Je spôsobená veľkými rozdielmi a nevyváženosťou systému. Pri použití tohto odhadovaného času pre dynamickú simuláciu však prejdú strediskom všetky prístroje a teda užívateľ nemusí vykonávať novú simuláciu pre zistenie celkového času prechodu prístrojov.

Druhý typ extrémnej odchýlky je menej častý, avšak taktiež k nemu dochádza, pri určitom manuálnom rozhodnutí technikov a bez použitia optimalizačných algoritmov. Pri tomto type odhad ukazuje konkrétny čas prechodu, lenže na niektorom úseku nastane veľké hromadenie prístrojov a simulácia nebude dokončená prechodom všetkých prístrojov, ani pri výraznom navýšení simulačného času. Tento problém taktiež dynamická simulácia chodu strediska odhalí a je možné ho odstrániť automatizovane, použitím optimalizačných algoritmov.



### **Algoritmy pre výpočet odhadu**

Algoritmy pre odhad času prechodu sa skladajú zo súčtu dvoch častí. Prvá časť je priemerný čas prechodu jedného prístroja strediskom. Jedná sa o výpočet počiatočnej podmienky, ktorá popisuje stredisko, v čase, keď nie je plne obsadené prístrojmi. Je to vlastne súčet priemerných časov úsekov a priemerných časov automatizovaných operácií počas kritickej cesty strediskom. Kritická cesta je cesta, ktorej zmena má priamy vplyv na celkový čas. Vychádza z teórie o metóde kritickej cesty, známej pod anglickým názvom *Critical Path Method* – CPM. Z pravidla sa jedná o najdlhšiu cestu [12]. Aplikácia taktiež ponúka možnosť náhľadu na kritickú cestu pri skúmaní odozvy strediska pri konkrétnom rozmiestnení technikov na prechod prístrojom. Toto skúmanie odozvy je však skôr pre analytické účely a nejedná sa o hlavnú funkcionálnosť aplikácie. Nastavenie záložky pre skúmanie odozvy a zobrazenie kritickej cesty popisuje samotný manuál aplikácie. Kritická cesta sa však využíva na rôznych iných miestach, bez vedomia užívateľa, pri rôznych algoritmoch, napríklad algoritmoch pre tento výpočet.

Ďalšou časťou súčtu je výsledok násobenia najdlhšieho intervalu strediska s počtom prístrojov. Algoritmy nachádzajú najdlhší interval pomocou nájdenia linky s najvplyvnejšou váhou, pomocou poradia váh. Najdlhší interval slúži taktiež ako interval strediska, pretože pri plnej saturácii (saturáciou sa zaoberá podkapitola 4.1.4 Simulácia chodu strediska) strediskom nemôže prejsť viac prístrojov, ako prejde úsekom s najdlhším intervalom.

### **Porovnanie odhadov s reálnym priebehom**

Pri optimalizovanom rozložení technikov v ukázkovom modeli obsahujúcom 250 prístrojov na vstupe (priložený, ako súbor v prílohe pod názvom `model_vsetci_tecnici_250_pristrojov_na_vstupe.modp`) je odhadovaný čas prechodu 57 hodín a 19 minút. Podrobná simulácia strediska však zistí, že reálny čas prechodu všetkých prístrojov strediskom je 59 hodín a 30 minút. Odchýlka medzi odhadovaným časom a skutočným je teda zhruba dve hodiny.

Pre navodenie extrémnej odchýlky je možné z tohto modelu odstrániť všetkých technikov začínajúcich na A B a C nastavením parametrov nedostupnosti (v prílohe súbor `model_tecnici_bez_ABC_250_pristrojov_na_vstupe.modp`). Takáto zmena značne rozhodí vyváženie strediska a odhad času bude značne skreslený. Odhad pre takto nastavený model je 116 hodín a 45 minút. Reálna simulácia však ukazuje, že prechod bude trvať len 83 hodín a 24 minút.

Spustením optimalizačných algoritmov sa rozloženie technikov opäť vyváži (v prílohe `model_tecnici_bez_ABC_250_pristrojov_na_vstupe_optimalizovany.modp`) a doba prechodu bude presnejšia. Odhad pre tento model je 69 hodín a 12 minút. Reálna dĺžka simulácie prechodu všetkých prístrojov je 70 hodín a 58 minút. Odhad je zas presnejší, s chybou len necelé dve hodiny.

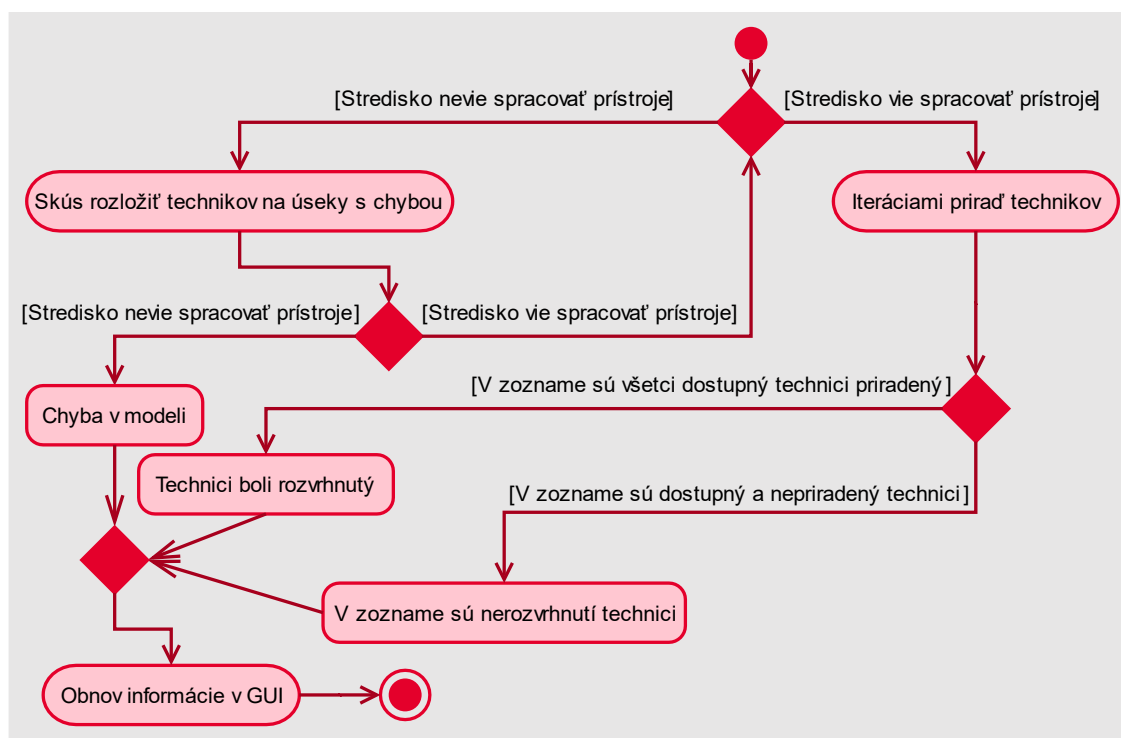
Z tohto porovnania vyplýva, podobný záver, ako z predchádzajúcich podkapitol. Algoritmy pre odhad fungujú s dobrou presnosťou (v príkladoch bola pri vyváženom stredisku odchýlka zhruba dve hodiny pri sedemdesiatich hodinách simulácie), avšak pri extrémnom narušení vyváženia technikov môže dôjsť k vysokej odchýlke (v príklade bola pre 83 hodín simulácie odchýlka vyše 33 hodín). Aj napriek týmto odchýlkam sú však algoritmy pre odhad času prechodu prístrojov strediskom výhodným nástrojom pre rýchly odhad potrebného vstupného času pre simuláciu. Podrobné skúmanie funkčnosti aplikácie a chodu strediska je rozobraté v podkapitole 4.1 Testovanie funkčnosti.

### 3.2.6 Automatizované priradovanie technikov

Automatizované priradovanie môže začať len za podmienky, že je daný model platný. Platnosťou modelu je myslené rozloženie technikov tak, že stredisko bude schopné spracovať všetky zadané prístroje a teda sa nestane, aby nejaký úsek nevedel spracovať nejaký prístroj, ktorý by mal a teda by vznikla neriešiteľná situácia. Validita modelu sa testuje pri každom spustení algoritmov

- Priradovacích a optimalizačných
- Simulačných, pre skúmanie odozvy a pre spustenie dynamickej simulácie

Pokiaľ tento model nie je platný pri automatizovanom priradovaní technikov, tak sa pred zahájením priradovacieho algoritmu spustí algoritmus, ktorý vyskúša doplniť vhodných technikov na neplatné úseky a vytvoriť funkčný model. Pokiaľ sa algoritmu podarí vytvoriť funkčný model spustia sa priradovacie algoritmy, pokiaľ nie, užívateľské rozhranie oznámi túto skutočnosť užívateľovi, ktorý sa môže pokúsiť o manuálnu korekciu. Tento postup pre automatizované priradenie technikov je popísaný diagramom aktivít nižšie (Obrázok 21).

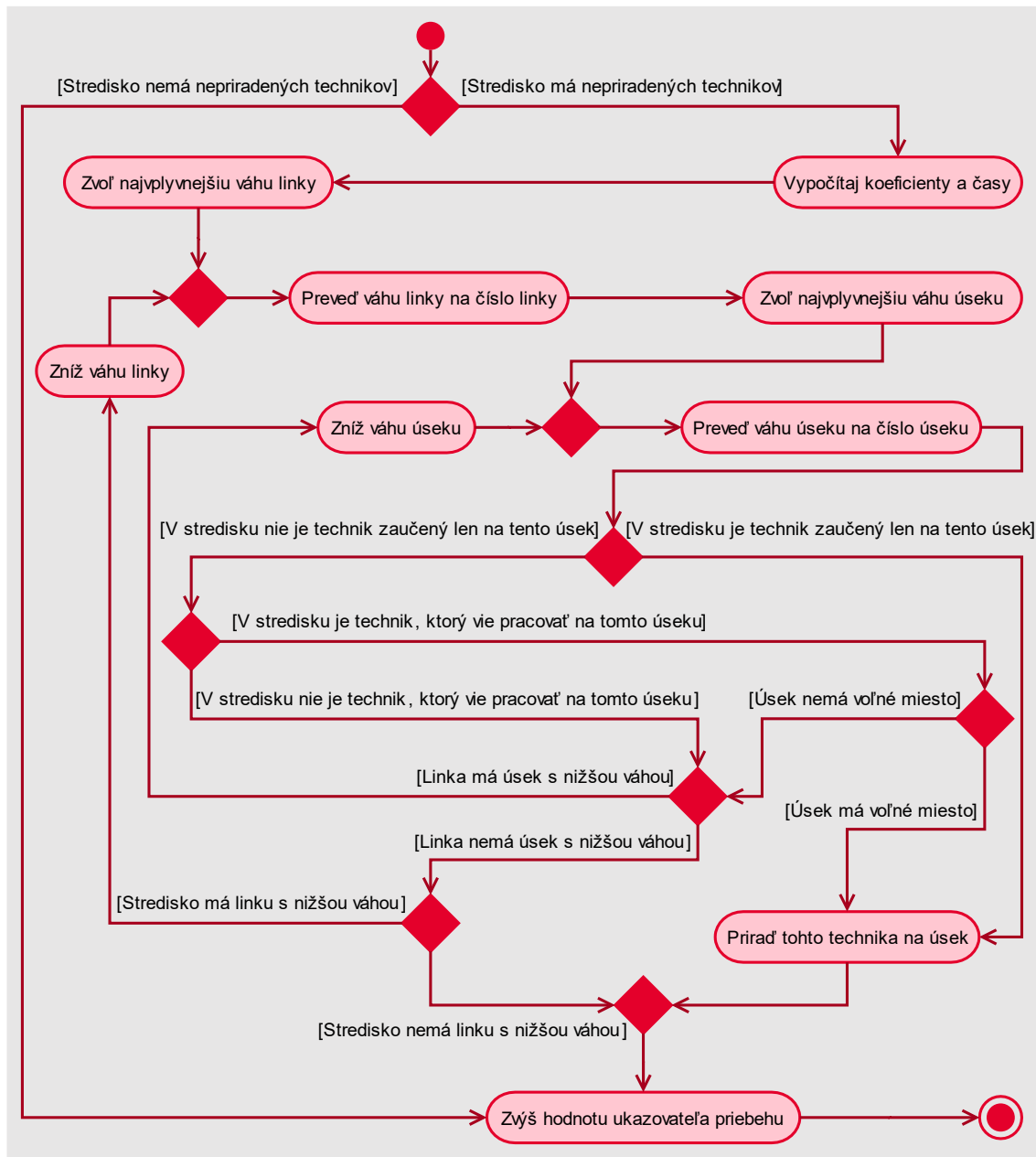


Obrázok 21 Zjednodušený diagram aktivít aktivovania automatizovaného priradovania technikov

Diagram vyššie taktiež poukazuje na dve možnosti ukončenia algoritmu pre automatizované priradovanie technikov. Pokiaľ algoritmus priradil všetkých dostupných technikov, užívateľské rozhranie oznámi užívateľovi tento stav, pokiaľ nie, užívateľské rozhranie na túto skutočnosť užívateľa upozorní. Diagram je zjednodušený, obnovovanie informácií v GUI zahŕňa okrem ohlásenia stavu užívateľovi aj prepočet koeficientov a ich prepísanie a taktiež upozornenie užívateľa na vykonanú zmenu zmenou farby pozadia. Užívateľ potom vie, že bola štruktúra strediska zmenená a je potrebné uložiť zmeny.

## Algoritmy rozvrhnutia technikov

Algoritmy, ktoré sa skrývajú za aktivitou „Iteráciami prirad' technikov“ na diagrame v predchádzajúcej podkapitole (Obrázok 21) je možné zjednodušiť popísať diagramom nižšie (Obrázok 22).



Obrázok 22 Zjednodušený diagram aktivít jednej iterácie priradenia technikov

Tento optimalizačne – prirad'ovací algoritmus je najjednoduchší zo všetkých ostatných optimalizačných algoritmov. Jeho diagram aktivít (Obrázok 22) zobrazuje jednu iteráciu priradenia jedného technika na úsek. Optimalizácia prirad'ovania cieľi na jednoduchú hrubú optimalizáciu pre rýchlosť prechodu prístroja strediskom. Cieľom tohto priradenia je zmenšiť vplyv najvplyvnejších liniek priradením technikov a teda vyvážiť rozloženie pracovnej sily. Algoritmy nepracujú s premiestňovaním technikov

podľa výkonnosti, preto sú jednoduché a poskytujú len hrubé rozmiestnenie technikov. Pri skúšaní rôznych vzoriek dát technikov vzišla požiadavka pre prednostné priradovanie technikov, ktorí dokážu pracovať len na jednom konkrétnom úseku. V prípade, že by táto požiadavka nebol splnená, mohli by byť na úsek primárne priradení technici, ktorí vedia pracovať aj na iných úsekoch a nakoniec by k nim boli aj tak pridaní spomínaní technici zaškolení len na tento úsek, aj pokiaľ by bola váha úseku malá.

Prvá aktivita je výpočet koeficientov a časov. V tejto aktivite dôjde k prepočtu všetkých kvantifikátorov liniek a úsekov. Kvantifikátory spolu s ich výpočtami sú podrobnejšie popísané v podkapitole 3.2.4 Kvantifikovanie liniek a úsekov. Prepočet na začiatku každej iterácie zaručí, že algoritmy budú pracovať so vždy aktuálnou kvantifikáciou liniek a úsekov. Po každom priradení technika totižto dochádza k zmenám kvantifikátorov, čo častokrát môže viesť až k zmene samotných priorít liniek a úsekov. Tento prepočet zabezpečuje metóda *vypocitajCasyKoefficienty* z knižnice optimalizácie algoritmov. Metóda potrebuje k výpočtom preposlať v parametroch tri databázy – databázu strediska, technikov a automatizovaných operácií. Volanie metódy pre získanie aktuálnych kvantifikátorov strediska je často využívané aj pri ostatných optimalizačných algoritmoch aplikácie.

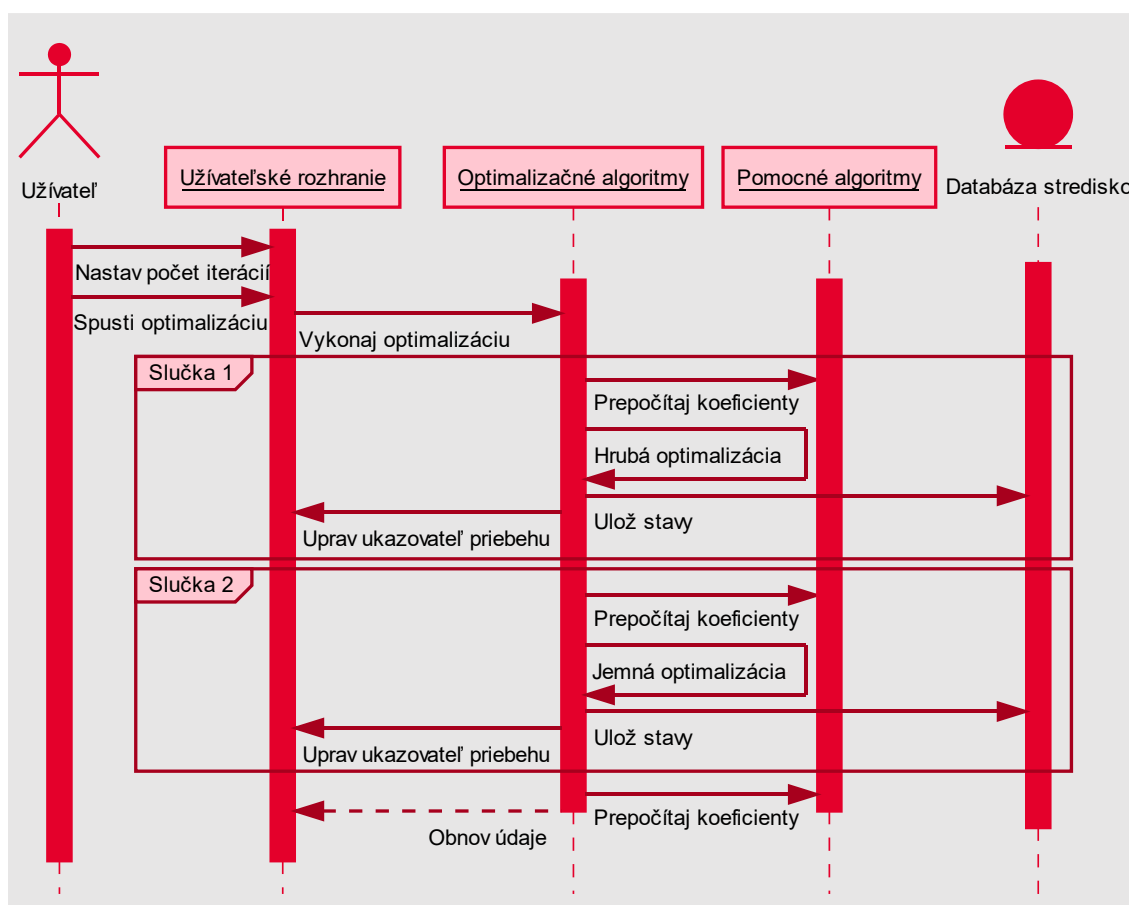
V diagrame sa nachádzajú dve aktivity pre prevod poradia váhy na identifikačné číslo v štruktúre. Výhodou používania poradia váh oproti identifikačným číslam je ten, že je možné krokovať podľa poradového čísla váhy a tým pádom prechádzať linky a úseky zoradené podľa veľkosti vplyvu. Metódy na prepočet týchto váh na číslo v štruktúre sú definované v knižnici optimalizačných algoritmov s názvami *cisloLinkySVahou* a *cisloOperacieSVahou* (Pozn. pojmy operácia a úsek sú pre tento prípad navzájom zameniteľné, operácia je proces vykonávaný na úseku. Pri terminológii v dátovej štruktúre modelu strediska je možné používať taktiež linku zloženú namiesto z úsekov z operácií. Diplomová práca však uprednostňuje jednotný pojem úsek, pre lepšiu prehľadnosť, avšak niektoré metódy v zdrojovom kóde obsahujú názov operácia). Parametre metód sú poradie váhy a v prípade metódy *cisloOperacieSVahou* aj číslo linky. Návrátová hodnota metód je typu *int* a vyjadruje poradové číslo linky v štruktúre strediska, prípadne poradové číslo úseku vo vektore linky. Metóda *cisloLinkySVahou* je založená na prechádzaní liniek v stredisku a porovnávaní hodnoty poradia váhy s poradím uloženým v štruktúre. Keď metóda nájde linku s rovnakým poradím váhy vráti jej poradové číslo. Metóda *cisloOperacieSVahou* funguje rovnako, akurát prehľadáva úseky v konkrétnej linke vymedzenej parametrom.

Aktivitu pridanie technika poskytuje metóda triedy databázy strediska *assignTechnik*, ktorej parametrami sú číslo linky, číslo operácie a meno technika. Metóda pozostáva len z jedného riadku kódu, ktorý priamo na konkrétnu pozíciu pridáva technika do vektora strediska. Čísla sú čísla poradia vo vektorovej štruktúre typu *int* a meno je textový reťazec. Pred pridaním algoritmy najskôr testujú, či má úsek voľné miesto. Na zistenie je použitá metóda *getMaxPocetTech*, taktiež z databázy strediska.

Pre vyhľadávanie nepriradených a zároveň dostupných technikov slúži metóda *volnyTechnikPreOp* implementovanej v triede databázy technikov. Parametrom tejto metódy je textový reťazec názvu operácie a návratová hodnota je textový reťazec mena technika. Pre vyhľadanie technika, ktorý je zaučený len na jeden úsek slúži metóda *volnyTechnikKtoryVielbaOp*. Táto metóda je podobná metóde *volnyTechnikPreOp*, akurát navyše kontroluje, či má technik v databáze pridaný len jeden úsek.

### 3.2.7 Algoritmy pre optimalizáciu rozloženia technikov

Tieto algoritmy sú založené na presúvaní priradených technikov. Skladajú sa z dvoch hlavných optimalizačných blokov. Oba bloky sú zabalené v slučkách, ktorých počet opakovaní nastavuje užívateľ pomocou vloženia číselného údaju do príslušného textového poľa. Prvý blok vykonáva hrubú optimalizáciu založenú na presúvaní vhodných technikov z vhodných operácií s nižším vplyvom na vhodné operácie s vyšším vplyvom. Druhý blok obsahuje algoritmy pre jemnú optimalizáciu. Jemná optimalizácia spočíva vo vzájomnej výmene technikov medzi operáciami na základe ich výkonnosti. Sekvenčný diagram interakcie algoritmov je znázornený na obrázku nižšie (Obrázok 23). Tento diagram je značne zjednodušený, zobrazuje interakciu na vyššej abstraktnej úrovni.



Obrázok 23 Zjednodušený sekvenčný diagram interakcie algoritmu optimalizácie

Vyššie uvedený diagram (Obrázok 23) v prvom kroku poukazuje potrebnú interakciu užívateľa pre zadanie počtov iterácií. V aplikácií sú v poliach prednastavené hodnoty, je však vhodné, aby užívateľ podľa rozsiahlosti svojho modelu toto číslo upravil. Ďalším krokom užívateľ stlačením tlačidla spúšťa optimalizačné algoritmy, stlačenie tlačidla taktiež vykoná automatickú kontrolu validity modelu. Pokiaľ je model platný, vykonajú sa optimalizačné algoritmy, pokiaľ je model neplatný, optimalizácia sa nevykoná a užívateľ je na túto skutočnosť upozornený.

Následne aplikácia vstúpi do prvej optimalizačnej slučky, v ktorej na začiatku prebehne prepočítanie koeficientov, pre získanie vždy aktuálnych kvantifikátorov pred každou iteráciou slučky. Toto prepočítanie využíva tri databázy, databázu strediska, technikov a automatizovaných operácií, avšak kvôli prehľadnosti diagramu neboli v diagrame uvedené. Najdôležitejšie kvantifikátory a ich výpočty sú podrobnejšie popísané v podkapitolách v časti 3.2.4 Kvantifikovanie liniek a úsekov.

V ďalšom kroku sa vykonávajú hrubé optimalizačné algoritmy pre presun technikov. Stav sa uloží do databázy strediska, podľa vzorca sa zvýši ukazovateľ priebehu a slučka ide od začiatku. Slučka pre jemnú optimalizáciu na tejto úrovni funguje podobne, akurát s inými optimalizačnými algoritmami. Ostatné kroky má však rovnaké s rovnakou funkcionalitou.

Po prevedení optimalizačných algoritmov sa nakoniec posledný krát prepočítajú kvantifikátory a spolu s nimi aj prepočet času prechodu prístrojov strediskom (3.2.5 Odhad času simulácie). Ako posledný krok sa vykoná obnovenie údajov v užívateľskom rozhraní. Do prvku zoznamu sa vypíšu sa aktuálne intervaly, koeficienty a váhy. Do časti s informáciami o modeli strediska sa vypíše aktuálny koeficient strediska spolu s odhadovanou dobou prechodu. Taktiež sa obnovia stavy pracovísk z mapy modelu do stromovej štruktúry.

### **Hrubá optimalizácia pre rýchlosť prechodu**

Algoritmy pre hrubú optimalizáciu sú jednoduchšie, ako algoritmy pre jemnú optimalizáciu. Ich princíp spočíva v prehľadávaní strediska od linky a úseku s najvyššou váhou postupne smerom dolu v snahe nájsť úsek, ktorému by bolo vhodné pridať technika. Zároveň popri tomto krokovaní dochádza taktiež ku krokovaniu od najnižšej váhy v snahe nájsť úsek vhodný na odobratie technika. Podobný princíp krokovania poradia váh liniek a úsekov od najvyššej váhy, je využívaný pri algoritme rozvrhovania technikov (Obrázok 22). Krokovanie liniek a úsekov podľa poradia váh a prevádzanie poradia váh na číslo umiestnenia v dátovej štruktúre je podrobnejšie rozpísané v podkapitole *Algoritmy rozvrhnutia technikov*.

Pokiaľ je pri krokovaní od najvplyvnejšej váhy na úseku voľné aspoň jedno miesto, algoritmy začnú prehľadávať stredisko z opačného konca, od linky a úseku s najnižšou váhou, smerom k vyšším váham v snahe nájsť úsek s nižším intervalom, ktorý by bol vhodný pre presun technika. Pre každý úsek algoritmus vypočíta odhad intervalu, ktorý by mal úsek po odobratí technika. Vzorec pre výpočet intervalu je nasledovný:

$$Interval_{po\ odobratí} = (k_{odobratia} + interval_{úseku}) * \frac{percentá\ linky}{100} \quad (3)$$

Interval je teda súčet koeficientu odobratia technika s priemerným výstupným intervalom úseku s nižšou váhou, ktorý je následne upravený korekciou. Výpočty týchto intervalov, koeficientov a informácie o korekciách sú uvedené v podkapitolách časti 3.2.4 Kvantifikovanie liniek a úsekov. Následne je tento vypočítaný interval porovnaný s priemerným intervalom najvplyvnejšej linky. Pokiaľ je interval tohto úseku aj po odobratí technika menší, ako interval najdlhšieho úseku, algoritmy pokračujú ďalej a prehľadáujú zoznam technikov na úseku.

Algoritmy technikov postupne testujú, či sú spôsobilý pre prácu na úseku s vyššou váhou. Pokiaľ algoritmus nájde takého technika, overí, či technik nie je manuálne uzamknutý na pozícii a otestuje, či je úsek schopný fungovať aj bez technika. Ak áno, zmaže ho z pôvodného úseku a priradí ho na úsek s vyššou váhou, čím zníži jeho interval.

Pokiaľ by však odobratie technika navýšilo interval úseku na vyšší, ako interval najvplyvnejšej linky, alebo algoritmus nenájde vhodného technika, algoritmus pokračuje ďalej v prehľadávaní strediska smerom k vplyvnejším váham.

Ak by algoritmus nenašiel žiaden úsek vhodný na presun technika, alebo by nenašiel žiadneho vhodného technika na presun, pokračoval by ďalej v zrýchľovaní liniek a úsekov s nižšími váhami.

Algoritmus väčšinou využíva známe metódy popísane v tejto práci, obsahuje však aj nové metódy. Najvýznamnejšími novými metódami sú metóda *jeTechnikZamknuty* a metóda *vieUsekBezTechnikaSpracovatPristroje*.

Metóda *jeTechnikZamknuty* je implementovaná v triede databázy technikov. Jej parametrom je textový reťazec mena technika. Návratová hodnota je binárna. Princíp algoritmu metódy je v prechádzaní záznamov za účelom nájdenia záznamu konkrétneho technika v dátovej štruktúre a vrátenie binárnej hodnoty zámku na tejto pozícii.

Metóda *vieUsekBezTechnikaSpracovatPristroje* vracia binárnu hodnotu, ktorá vyjadruje, či je daný úsek platný aj po odobratí technika. Pokiaľ by tento úsek nebol platný, spôsobil by neplatnosť celého modelu a nefunkčnosť ostatných algoritmov. Táto metóda je definovaná v knižnici optimalizačných algoritmov. Pre svoju funkčnosť potrebuje veľký počet parametrov, okrem čísla linky, čísla úseku a textového reťazca mena technika je potrebná aj databáza strediska a databáza technikov. Z databázy strediska metóda načítava zoznam prístrojov, ktoré strediskom prejdú a taktiež aj zoznam technikov na úseku. Následne pomocou metód poskytnutých v databáze technikov metóda overí, či úsek dokáže spracovať prístroje aj bez konkrétneho technika a na základe toho vráti binárnu hodnotu.

### **Jemná optimalizácia pre rýchlosť prechodu**

Jemná optimalizácia podobne ako hrubá optimalizácia prechádza strediskom pomocou krokovania váh. Pre úsek s dlhším intervalom od linky a úseku s najvplyvnejšou váhou nižšie a pre úsek s menším intervalom od linky a úseku s najmenšou hodnotou váhy vyššie. Na rozdiel od hrubej optimalizácie, jemná optimalizácia pracuje na princípe vzájomnej výmeny technikov podľa ich výkonnosti. Snahou je výmena pomalšieho technika z úseku s dlhším intervalom za rýchlejšieho technika v úseku s kratším intervalom. Toto tvrdenie je však značne zjednodušené a je treba počítať aj s ďalšími inými parametrami.

Samotné algoritmy sú však v porovnaní s hrubou optimalizáciou výrazne zložitejšie. Pre každý dlhší úsek algoritmy skúšajú prechádzať technikmi na tomto úseku. Pre každého technika na úseku s väčším intervalom algoritmy skúšajú, či tento technik nie je zamknutý a či dokáže pracovať na kratšom úseku. Pokiaľ technik prejde týmito požiadavkami, algoritmy začnú hľadať rýchlejšieho technika na výmenu v úseku s kratším intervalom. Prechádzaním zoznamu technikov na úseku s kratším intervalom skúšajú technikov pre podobné podmienky – či technik dokáže pracovať na úseku s dlhším intervalom a či nie je manuálne zamknutý na pozícii.

Keď algoritmy nájdu dvoch vhodných technikov, nastáva ďalšia podmienka. Výmena technikov môže nastať len za podmienky, že sa nenaruší validita žiadneho z úsekov. K overeniu slúži metóda *vieUsekSoZamenouTechnikaSpracovatPristroje*. Parametrami metódy sú čísla úseku a linky, textové reťazce s menami technikov, technika na pridanie na úsek, meno technika pre odobratie z úseku a databázy strediska

a technikov. Algoritmy tejto metódy fungujú podobne, ako algoritmy metódy *vieUsekBezTechnikaSpracovatPristroje*, akurát po odobratí technika sa taktiež k technikom, ktorý ostali skúsi pridať technik z druhého úseku. Totižto úsek po odobratí technika nemusí vedieť spracovať niektoré prístroje, ale po následnom pridaní technika z druhého úseku už bude vedieť spracovať všetky potrebné prístroje. Metóda pre otestovanie validity úseku po odobratí technika je popísaná v predchádzajúcej podkapitole *Hrubá optimalizácia pre rýchlosť prechodu*.

Pokiaľ je možné technikov zameniť bez narušenia validity niektorého z úsekov vypočíta sa priemerný čas oboch technikov pre úsek s dlhším intervalom. Počítanie priemerného času pre technika je popísané v podkapitole *Algoritmy pre výpočet intervalov*. Ak je priemerný čas technika z úseku s kratším intervalom pre úsek s dlhším intervalom kratší ako priemerný čas pôvodného technika na tomto úseku, tak sa technici zamenia a algoritmy pokračujú na ďalšiu iteráciu. Pokiaľ je však výhodnejšie nechať pôvodného technika na úseku s dlhším intervalom pokračujú algoritmy prechádzaním ďalších technikov. V prípade, že algoritmy nenájdu vhodných technikov na výmenu pokračujú po úsekoch a linkách ďalej.

V tomto type optimalizácie sa nepoužíva výpočet s intervalmi po odobratí technika, pretože k tak veľkej zmene intervalu nedôjde. Algoritmus technikov zamení, úseku s kratším intervalom sa tento interval po zámene môže predĺžiť, nie však tak razantne, ako keby sa technik odobral úplne.

### 3.2.8 Simulácia chodu strediska

Simulačná časť je najzložitejšia časť aplikácie. Obsahuje najkomplexnejšie algoritmy a volá dlhé metódy. Diagram interakcií je uvedený na obrázku nižšie (Obrázok 24). Užívateľ zadá požadovanú dĺžku simulácie a samotnú simuláciu spustí stlačením tlačidla. Pred spustením simulácie môže taktiež užívateľ zmeniť názov simulácie, ale tento krok nie je nutný. Názov simulácie môže ostať aj na prednastavenej hodnote. Na vykonávanie simulácie to nemá žiaden vplyv.

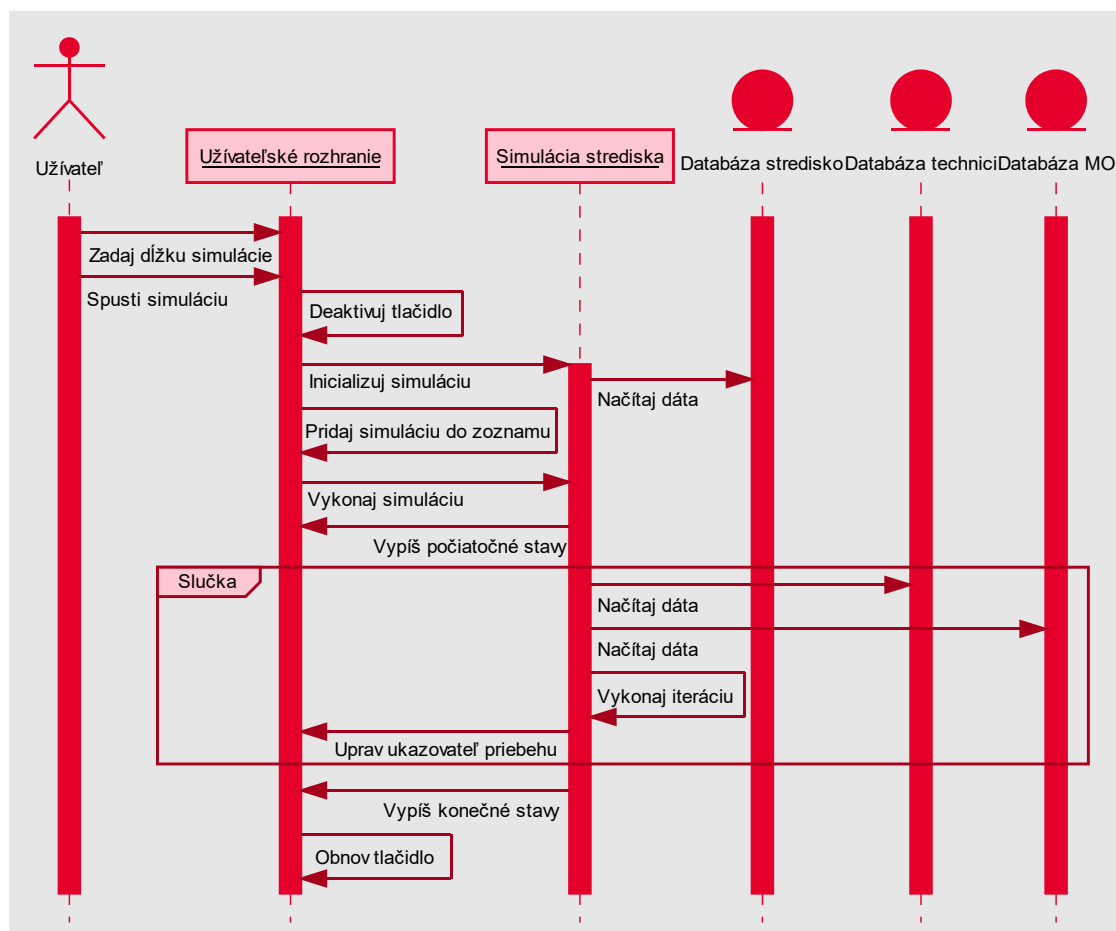
Simulácia trvá dlhší čas (rádovo desiatky sekúnd), z toho dôvodu ihneď po stlačení tlačidla pre začatie simulácie sa toto tlačidlo deaktivuje, aby užívateľ nemohol opätovne stlačiť tlačidlo počas behu simulácie. Narušilo by to priebeh simulácie a mohlo by dôjsť až k pádu aplikácie. Toto tlačidlo sa stane znova aktívnym po prebehnutí simulácie.

V prvom kroku nastane inicializácia dátovej štruktúry simulácie (podkapitola *Dátová štruktúra simulácie*). Inicializačné algoritmy používajú pre vytvorenie štruktúry databázu strediska, v ktorej sú uložené informácie o stredisku, napríklad model kostry strediska, rozloženie technikov a prístrojov na úsekoch, kroky operácií a ďalšie parametre.

Ďalším krokom dôjde v užívateľskom rozhraní k pridaní tejto simulácie do zoznamu simulácií v časti užívateľského rozhrania pre výber prebehnutej simulácie. V tejto ďalšej časti rozhrania pre výber simulácie užívateľ po vykonaní simulačných algoritmov môže vybrať simuláciu pre podrobnejšiu analýzu. Táto možnosť je v aplikácii zahrnutá z čisto praktického dôvodu. Umožňuje porovnávanie výsledkov starších simulácií, bez nutnosti opätovného simulovania už odsimulovanej simulácie. Je to určitý kompromis šetrenia času na úkor vyťaženia operačnej pamäte. Pokiaľ by však aplikácia mala veľké simulácie a užívateľ by vyhodnotil, že zaberá príliš veľa miesta v operačnej pamäti, je tu taktiež



možnosť zmazať vybranú simuláciu.



Obrázok 24 Zjednodušený sekvenčný diagram interakcií pri spustení simulácie

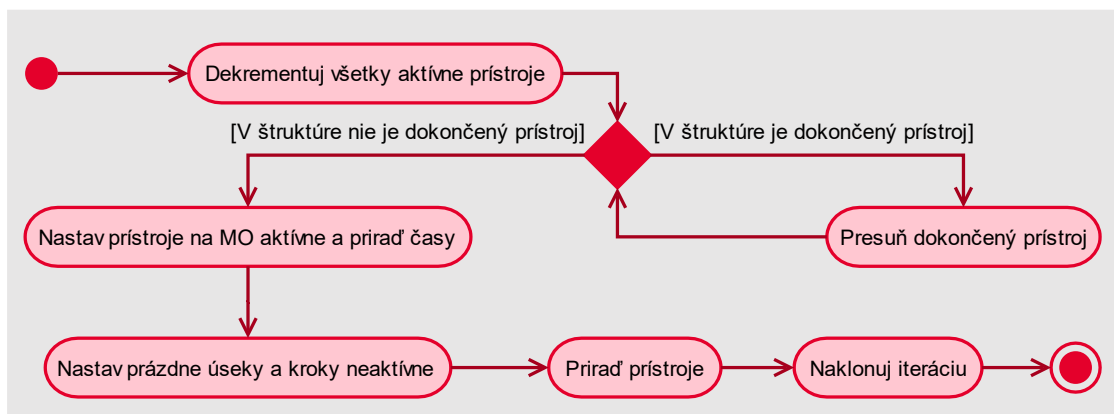
Po pridaní názvu simulácie do zoznamu simulácií dôjde z objektu obsluhujúceho užívateľské rozhranie k volaniu metódy pre vykonanie simulácie v objekte simulácie strediska. Metóda na začiatku vypíše počiatočné stavy strediska do prvku textového poľa v užívateľskom rozhraní. Pre možnosť rýchleho porovnania stavov strediska pred spustením simulácie a po ukončení simulácie, teda po ubehnutí doby, ktorá bola v simulácii nastavená, po ukončení simulačných algoritmov taktiež dochádza k výpisu konečných stavov strediska do podobného prvku textového poľa.

V samotnej simulačnej slučke dochádza k spúšťaniu zložitého algoritmu pre vykonávanie iterácií. Algoritmus pracuje so štruktúrou simulácie strediska (Obrázok 26) a k svojmu chodu potrebuje databázy technikov a automatizovaných operácií. Samotnú databázu strediska však už algoritmus po inicializácii ďalej nevyužíva, simulačná štruktúra strediska sa značne líši od štruktúry strediska v databáze strediska. Simulačná štruktúra je podrobnejšie rozobraná v podkapitole *Dátová štruktúra simulácie*.

Popri vykonávaní slučky, každou jej iteráciou objekt simulácie strediska volá metódu prvku ukazovateľa priebehu pomocou objektu užívateľského rozhrania. Toto volanie slúži na úpravu hodnoty ukazovateľa priebehu pre informovanie užívateľa o priebehu simulácie. Viac informácií o prvku ukazovateľ priebehu sa nachádza v podkapitole *Ostatné prvky*.

### Zjednodušený celkový pohľad na iteráciu simulácie

Samotnú iteráciu simulácie je možné rozdeliť na menšie časti, podľa funkčných celkov, ktoré sa v danej časti vykonávajú. Tieto celky sú vlastne aktivity a z toho dôvodu je pre takéto rozdelenie vhodným popisným nástrojom diagram aktivít. Diagram pre iteráciu simulácie je na obrázku nižšie.



Obrázok 25 Zjednodušený diagram aktivít iterácie simulácie

Táto vizualizácia (Obrázok 25) poskytuje vysokoúrovňový náhľad na funkcionality iterácie. Pre každú aktivitu tohto diagramu je zvyčajne volaná jedna metóda. Každá metóda má rozličnú komplikovanosť a taktiež počet volaní ďalších pomocných metód. Rozhodovacia štruktúra je taktiež založená na volaní metódy. Na základe jej návratovej hodnoty následne nastáva rozhodovanie. Všetky metódy volané z tejto abstraktnej úrovne sú definované v knižnici simulačných algoritmov.

Prvá volaná metóda je pre dekrementovanie aktívnych prístrojov je najjednoduchšia v porovnaní s ostatnými metódami a slúži výhradne na dekrementáciu počítadiel času všetkých aktívnych prístrojov. Aktívny prístroj je prístroj na ktorom technik práve pracuje, prípadne prístroj, na ktorom sa vykonáva nejaký automatizovaný proces. Pokiaľ je prístroj na nejakom kroku aktívny, obsahuje jeho vektor v binárnej premennej s názvom čaká hodnotu *false*. Každý takýto prístroj má vo svojom vektore tiež aj číselnú premennú s počtom iterácií, ktoré vyjadrujú ostávajúci čas do dokončenia kroku. Metóda pre dekrementáciu tejto premennej sa volá *dekrementujVsetkyAktivnePristroje*. Jej parametrom je vektor liniek strediska z aktuálnej štruktúry simulácie strediska pre konkrétnu iteráciu. Principiálne funguje na prechádzaní celej tejto štruktúry až na úrovni samotných krokov a na každom kroku prechádza prístroje, pri čom každý prístroj testuje na aktivitu. Pokiaľ je prístroj aktívny, dekrementuje počítadlo ostávajúcich iterácií a pokračuje hľadaním ďalšieho aktívneho prístroja. Jej jednoduchosť je možné zdôrazniť tým, že na úrovni zdrojového kódu sa skladá len z piatich *for* cyklov a iba jednej *if* podmienky, ktorá v sebe obsahuje len jeden príkaz.

Podobne jednoduchá metóda je použitá v podmienke, ktorá tvorí slučku, pokiaľ sa v stredisku nachádza nejaký dokončený prístroj. Taktiež prehľadáva stredisko a testuje prístroje. Funguje však mierne odlišne. Pre funkčnosť podmienky táto metóda vracia binárnu hodnotu, či nejaký dokončený prístroj našla, alebo nie. Okrem tohto sú jej parametrami okrem vektoru liniek, podobne ako pri metóde v predchádzajúcom odstavci aj pomocou referencie posielané číselné hodnoty súradníc pre prístroj. Súradnice označujú pozíciu prístroja v stredisku. Metóda ich hodnoty upravuje pre ďalšie použitie.

Pokiaľ je v štruktúre strediska dokončený prístroj, spustia sa algoritmy aktivity presunu dokončeného prístroja. K lokalizácii dokončeného prístroja v dátovej štruktúre simulácie strediska slúžia číselné premenné označujúce pozíciu prístroja v stredisku, čísla linky, úseku, operácie, kroku a prístroja modifikované metódou z predchádzajúceho odstavcu. Algoritmy pre presun dokončeného prístroja sú zložitejšie a je im venovaná samostatná podkapitola *Presúvanie dokončeného prístroja*. Pokiaľ nebol prístroj presunom dokončený nastaví sa mu parameter čaká na hodnotu *true*.

Ak štruktúra strediska neobsahuje žiadny dokončený prístroj, pokračuje algoritmus ďalšou aktivitou. V tejto aktivite sa nastaví prístroje, ktoré boli presunuté na automatizované procesy. Využíva sa tu metóda *NastavMOAktivnePridajCasy*, ktorá prehľadá dátovú štruktúru simulácie na prístroje, ktoré boli presunuté na automatizované úseky v predchádzajúcom kroku. Vyhľadá teda prístroje, ktoré sú na automatizovaných krokoch a majú parameter čaká. Pokiaľ metóda nájde takéto prístroje, nastaví im počítadlo iterácií podľa času načítaného z databázy medzi – operácií a aktivuje prácu na prístroji nastavením hodnoty parametru čaká na *false*.

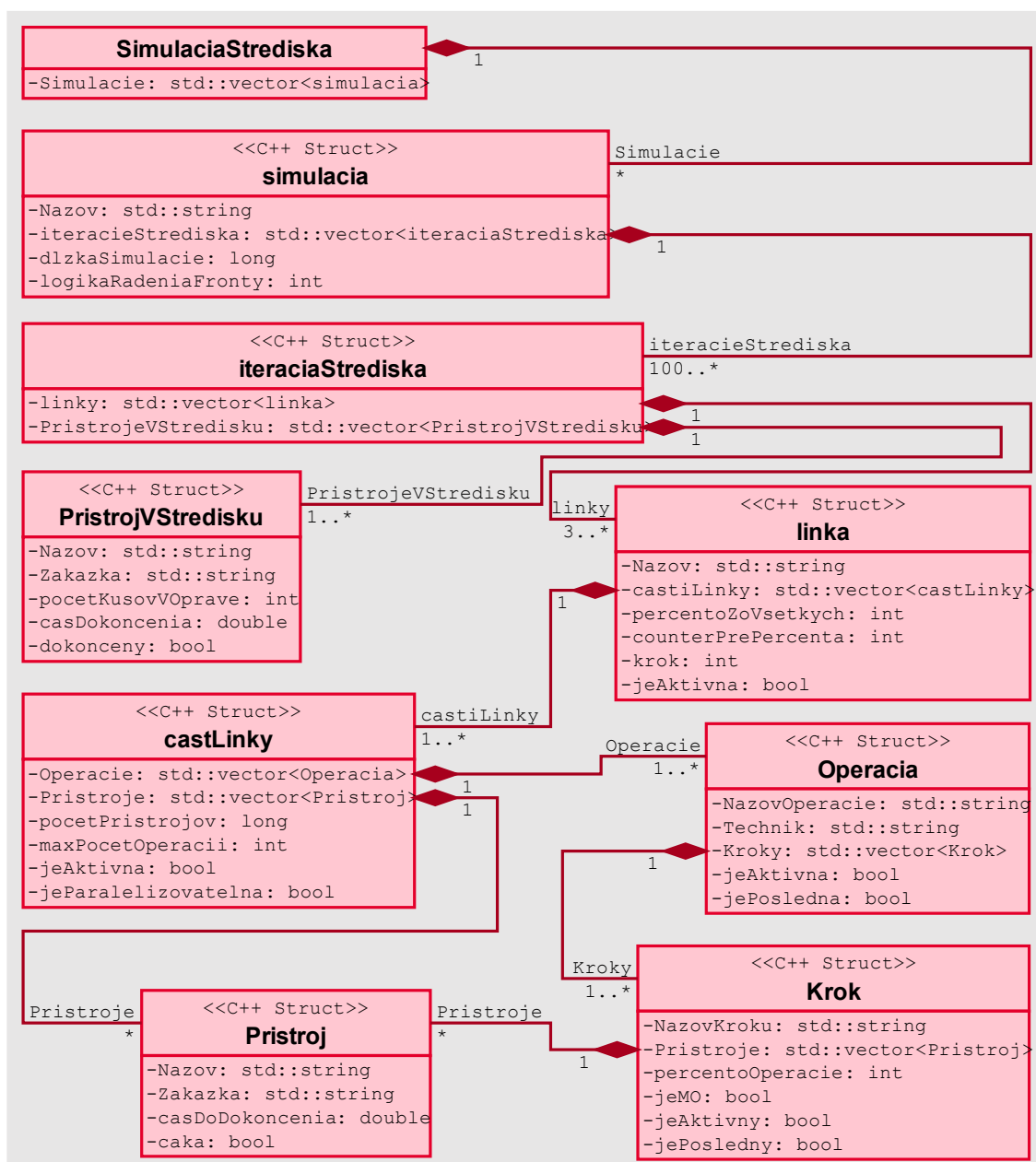
Po presunutí dokončených prístrojov z krokov ostali niektoré kroky bez aktívnych prístrojov. Logicky sa teda musí vykonať aktivita, ktorá tieto kroky nejak poznačí, aby mohli byť jednoducho nájdené algoritmi pre priradovanie prístrojov nevyužitým technikom. Algoritmy tejto aktivity sú taktiež zabalené v metóde, pre zvýšenie prehľadnosti kódu. Tieto algoritmy sú už zložitejšie, najskôr prehľadávajú dátovú štruktúru a označujú samotné neaktívne kroky. Prehľadávanie dátovej štruktúry je zložené z vnorených slučiek zahŕňajúcich prehľadávanie liniek, na nich úsekoch, v nich zas prehľadávanie operácií, ich krokov a nakoniec samotných prístrojoch. Tieto vnorené slučky sú však pre zmenšenie veľkosti zdrojového kódu využité aj na označovanie neaktívnych operácií, úsekov a liniek. Po určení krokov v operácií sa nastaví príznak aktivity samotnej operácie. Pokiaľ operácia nemá žiaden aktívny na technikovi závislý krok nastaví sa príznak neaktivity samotnej operácií. Takto slučky pokračujú ďalej, po nastavení všetkých operácií sa nastavujú jednotlivé úseky a po nastavení úsekov dochádza k nastavovaniu samotných liniek.

Následne sa vykoná aktivita priradovania prístrojov technikom, ktorí nepracujú na žiadnom prístroji, či už z dôvodu dokončenia práce na prístroji, alebo zahájením automatizovaného procesu. K vykonaní tejto aktivity sa používa metóda s názvom *PriradPristroje*. Podobne ako do iných metód v iterácii simulácie sa jej parametrom posielajú časť dátovej štruktúry obsahujúcej vektor liniek. Ďalším preposielaným parametrom je vektor prístrojov v stredisku. Tento vektor sa využíva pri dokončovacej linke na zistenie, či na dokončovaciu linku prišli všetky kusy prístrojov a je teda možné začať prístroj skladať dokopy. Posledným významným preposielaným parametrom je objekt databázy technikov. Táto databáza sa využíva pri zisťovaní času, ktorý konkrétny technik potrebuje na konkrétnom úseku pre konkrétny prístroj. Algoritmy tejto metódy sú pomerne zložité a je im venovaná samostatná podkapitola *Priradzovanie prístrojov nevyťažným technikom*. Výsledkom je však štruktúra s finálnymi stavmi po vykonaní jednej iterácie.

V poslednej aktivite diagramu nastáva naklonovanie dátovej štruktúry tejto iterácie na ďalšiu pozíciu, ktorá je spracovávaná ďalšou iteráciou. Obsluhuje ju metóda s názvom *naklonujIteráciu*, ktorej parametrom je celá dátová štruktúra s iteráciami a číslo iterácie. Algoritmy následne vytvoria nový záznam v štruktúre a skopírujú iteráciu.

## Dátová štruktúra simulácie

Aplikácia obsahuje skoro v každej triede aspoň jednu dátovú štruktúru zloženú z vektorov obsahujúcich štruktúry s ďalšími vektormi ďalších štruktúr. Dátová štruktúra simulácie (Obrázok 26) je však najrobustnejšia zo všetkých ostatných štruktúr. Štruktúra je robustná, pretože musí obsahovať všetky parametre potrebné pre vykonávanie simulácie. Záznam o simulácii, spolu s parametrami simulácie – názvom a dĺžkou simulácie. Ďalej musí obsahovať vektor s iteráciami, v ktorom bude obsiahnutý zoznam prístrojov v stredisku spolu s potrebnými parametrami a celý procesný model strediska. Procesný model strediska je na rozdiel od modelu strediska s parametrami úsekov (obsahuje len kostru strediska, podkapitola *Modelovanie kostry strediska* a nastavenie úsekov, podkapitola *Nastavenie úseku na mape modelu strediska*) zložitejší. Vytvára sa pri inicializácii simulácie (3.2.8 Simulácia chodu strediska a Obrázok 24).



Obrázok 26 Zjednodušené zobrazenie dátovej štruktúry simulácie pomocou diagramu triedy

Na zjednodušenom diagrame triedy zameranom na zobrazenie dátovej štruktúry simulácie (Obrázok 26) je možné vidieť osem štruktúr, ktoré sú vkladané do vektorov. Trieda simulácie strediska obsahuje jeden vektor, do ktorého sa vkladajú jednotlivé vykonávané simulácie. Táto kapitola popisuje najdôležitejšie premenné štruktúr použitých v simulácii. Niektoré premenné nie sú podrobnejšie vysvetľované, pokiaľ nemajú pre chod simulácie významný vplyv, alebo pokiaľ boli do štruktúry pridané len kvôli možnosti ďalšieho rozšírenia aplikácie a v súčasnej aplikácii nemajú uplatnenie. Nasledujúce odstavce popisujú jednotlivé štruktúry:

Štruktúra *simulacia* obsahuje všetky dáta konkrétnej simulácie, ktorú užívateľ spustil. Je v nej poznačený názov simulácie, dĺžka simulácie, logika radenia fronty a vektor iterácií strediska. Logika radenia fronty je experimentálna premenná pre možnosť ďalšieho rozširovania aplikácie. V stredisku rovnako ako počas simulácie platí radenie fronty FIFO (3.2.4 Kvantifikovanie liniek a úsekov).

Štruktúra *iteraciaStrediska* obsahuje dva vektory. Vektor *linky*, ktorý obsahuje štruktúry strediska a vektor *PristrojeVStredisku*, ktorý obsahuje zoznam prístrojov v stredisku spolu s ďalšími premennými.

Štruktúra *PristrojVStredisku* obsahuje textové reťazce s názvom prístroja a zákazkou, premennú typu *int* vyjadrujúcu počet kusov v oprave, premennú typu *double* vyjadrujúcu čas dokončenia a binárnu hodnotu s názvom dokončený. Počet kusov v oprave vyjadruje počet kusov, na koľko sa prístroj po prechode prípravnou linkou rozdelí po dobu akú budú tieto kusy v opravných linkách. Táto premenná má význam pre zistenie, či na linku pre dokončenie dorazili všetky kusy prístroja a linka teda môže začať spracovávať prístroj ďalej. Do času dokončenia sa zapíše čas, kedy opravený prístroj opustí stredisko. Premenná *dokonceny* taktiež slúži na poznačenie skutočnosti, že opravený prístroj opustil stredisko, jej význam je pre jednoduchšie sledovanie prístroja v stredisku algoritmom pri spracovaní dát.

Štruktúra *linka* obsahuje textový reťazec s názvom linky, vektor *castiLinky*, premenné typu *int* s názvami *percentoZoVsetkych*, *counterPrePercenta* a *krok*. Ďalej taktiež štruktúra obsahuje binárnu premennú vyjadrujúcu aktivitu linky. Vektor s časťami linky obsahuje ďalšiu časť štruktúry strediska. Premenná *percentoZoVsetkych* vyjadruje koľko percent zo všetkých prístrojov linkou prejde (podrobnejšie rozobraté v podkapitole *Modelovanie kostry strediska*). K nej patrí ďalšia premenná zaisťujúca funkčnosť distribúcie prístrojov *counterPrePercenta*. Obsahuje aktuálnu hodnotu počtu percent prístrojov z menších vzoriek, ktoré prešli predchádzajúcou linkou, aby bolo možné algoritmiami určiť, či má ďalší prístroj linkou prejsť, alebo nie. Premenná *krok* slúži na označenie danej linky ako prípravnej, opravnej, alebo dokončovacej. Samozrejme tento údaj je možné dopočítať, pre možnosť použitia jednoduchších algoritmov je tento údaj už vypočítaný v premennej. Binárna hodnota aktivity určuje, či linka obsahuje aspoň jeden aktívny úsek. Pokiaľ áno, znamená, že na linke robí aspoň jeden technik a linka je tým pádom aktívna. Pokiaľ na žiadnom z úsekov technik práve nepracuje, tak to znamená, že celá linka je neaktívna a parameter bude mať hodnotu *flase*. Samotné aktívne kroky automatizovaných operácií sa nepočítajú, ako aktivita pracovníkov na úseku, pretože sa vykonávajú nezávisle na technikoch a technik tým pádom nie je na linke potrebný a teda linka je neaktívna.

Štruktúra *castLinky* obsahuje dva vektory. Vektor s operáciou a vektor obsahujúci prístroje. Taktiež obsahuje číselné premenné *pocetPrístrojov*, *maxPocetOperacii* a binárne premenné *jeAktivna* a *jeParalelizovatelna*. Z týchto premenných sa využíva počet prístrojov a aktivita časti linky. Ostatné premenné sú pre prípadne rozšírenie aplikácie. Štruktúra *castLinky* vyjadruje úsek, ktorý je akýsi medzi krok medzi linkou a samotnou operáciou. Vektor s prístrojmi využíva štruktúry *Pristroj* ako zoznam prístrojov na úseku spolu s ich stavmi. Táto štruktúra je využívaná taktiež rovnomenným vektorom v štruktúre *Krok*. Štruktúra *Pristroj* je popísaná v poslednom odstavci, je však odlišná od štruktúry *PristrojVStredisku*. Vektor *Operacie* sa plní štruktúrou *Opercia*.

Štruktúra *Operacia* vyjadruje operáciu na úseku spolu s jej stavmi. Každý technik vykonáva svoju operáciu a teda vektor *Operacie* má presne toľko štruktúr operácie, koľko má úsek pridelených technikov. Štruktúra obsahuje textové reťazce s názvom operácie a menom vykonávajúceho technika. Binárne premenné má dve. Jednu s názvom *jeAktivna*, pre informáciu o tom, či technik na operácii práve pracuje a druhú s názvom *jePosledna*, ktorá značí, či je operácia poslednou operáciou na linke. Tejto premennej využívajú simulačné algoritmy v podmienkach, pre rýchle otestovanie pozície operácie v linke. Ďalej štruktúra *Operacia* obsahuje vektor *Koky*. Krok je na rozdiel od štruktúry modelu strediska typickým prvkom práve procesného modelu.

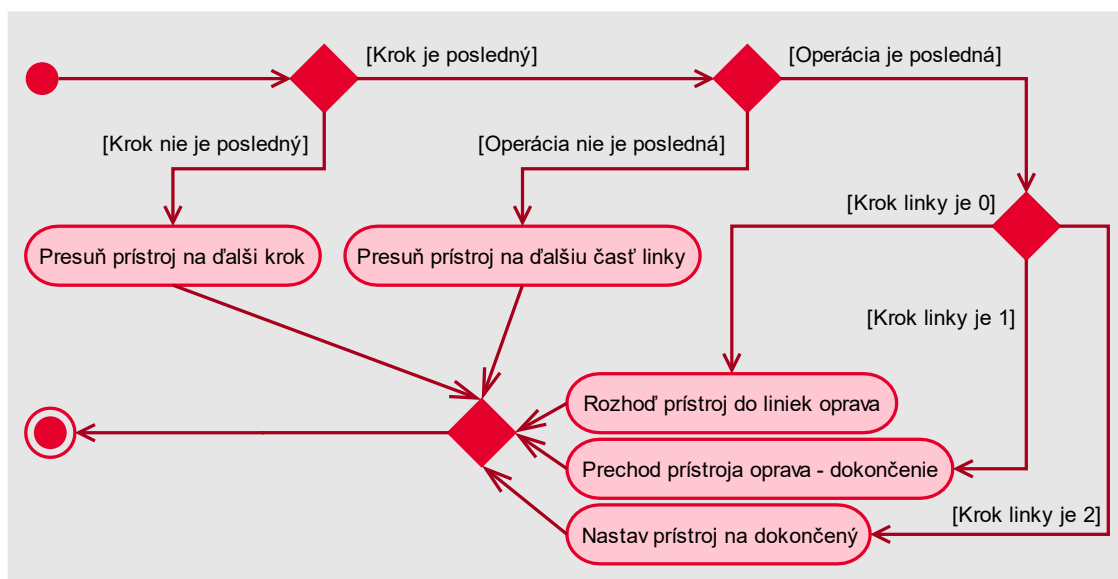
Štruktúra *Krok* obsahuje textový reťazec s názvom kroku, číselný údaj typu *int* s názvom *percentoOperacie* a tri binárne premenné – *jeMO*, *jeAktivny* a *jePosledny*. Okrem týchto premenných taktiež obsahuje vektor prístrojov, podobne, ako štruktúra *castLinky*. Premenná *percentoOperacie* vyjadruje koľko percent krok tvorí z celkového času technika na operácii. Premenné *jeAktivny* a *jePosledny* majú podobný význam ako podobné premenné v štruktúre *Operacia*. Hodnota premennej *jeMO* značí, či sa jedná v prípade hodnoty *true* o automatizovaný krok operácie, alebo v prípade hodnoty *false* o krok vykonávaný technikom.

Štruktúra *Pristroj* je zložená z aktuálnych stavov prístrojov na konkrétnych častiach linky, prípadne krokoch. Pre identifikáciu prístroja štruktúra obsahuje dva textové reťazce, jeden s názvom prístroja a druhý s názvom zákazky. Taktiež obsahuje číselnú premennú *casDoDokoncenia* vyjadrujúcu čas v iteráciách, ktorý prístroj musí na danom kroku ešte absolvovať, pokiaľ je aktívny. Binárna premenná s názvom *caka* popisuje aktivitu na prístroji. Pokiaľ má premenná hodnotu *true*, prístroj čaká vo fronte na časti linky, prípadne vo fronte kroku na spracovanie, pokiaľ má hodnotu *false*, tak práve tento prístroj technik, prípadne automatizovaný proces spracováva.

### **Presúvanie dokončeného prístroja**

Algoritmy pre presúvanie prístroja sú volané vždy len v prípade, že metóda pre prehľadávanie strediska na dokončený prístroj nájde nejaký konkrétny dokončený prístroj (*Zjednodušený celkový pohľad na iteráciu simulácie*). Algoritmy teda už poznajú umiestnenie prístroja v stredisku a na základe jeho umiestnenia vykonajú ďalšie kroky.

Algoritmus pre rozhodovanie využíva binárnych hodnôt *jePosledny*, *jePosledna* a číselných hodnôt *krok* zo štruktúry *linka* z dátovej štruktúry simulácie. Tieto premenné sú detailnejšie rozpísané v predchádzajúcej podkapitole *Dátová štruktúra simulácie*. Samotný algoritmus je zobrazený na diagrame na ďalšej strane (Obrázok 27).



Obrázok 27 Zjednodušený diagram aktivít pre presunutie jedného prístroja

Ako prvý algoritmus testuje, či je krok, na ktorom sa dokončený prístroj nachádza posledný. Pokiaľ krok posledný nie je, algoritmus len jednoducho presunie prístroj na ďalší krok. Pokiaľ je krok posledný, algoritmus otestuje, či je operácia, na ktorej sa krok nachádza posledná. Pokiaľ operácia nie je posledná, algoritmus taktiež len jednoducho presunie prístroj do fronty na ďalšiu časť linky. Ak je však operácia posledná algoritmus sa rozhodne, ktorú aktivitu vykoná na základe toho, či je daná linka, na ktorej sa krok nachádza linka pre prípravu, opravu, alebo dokončenie.

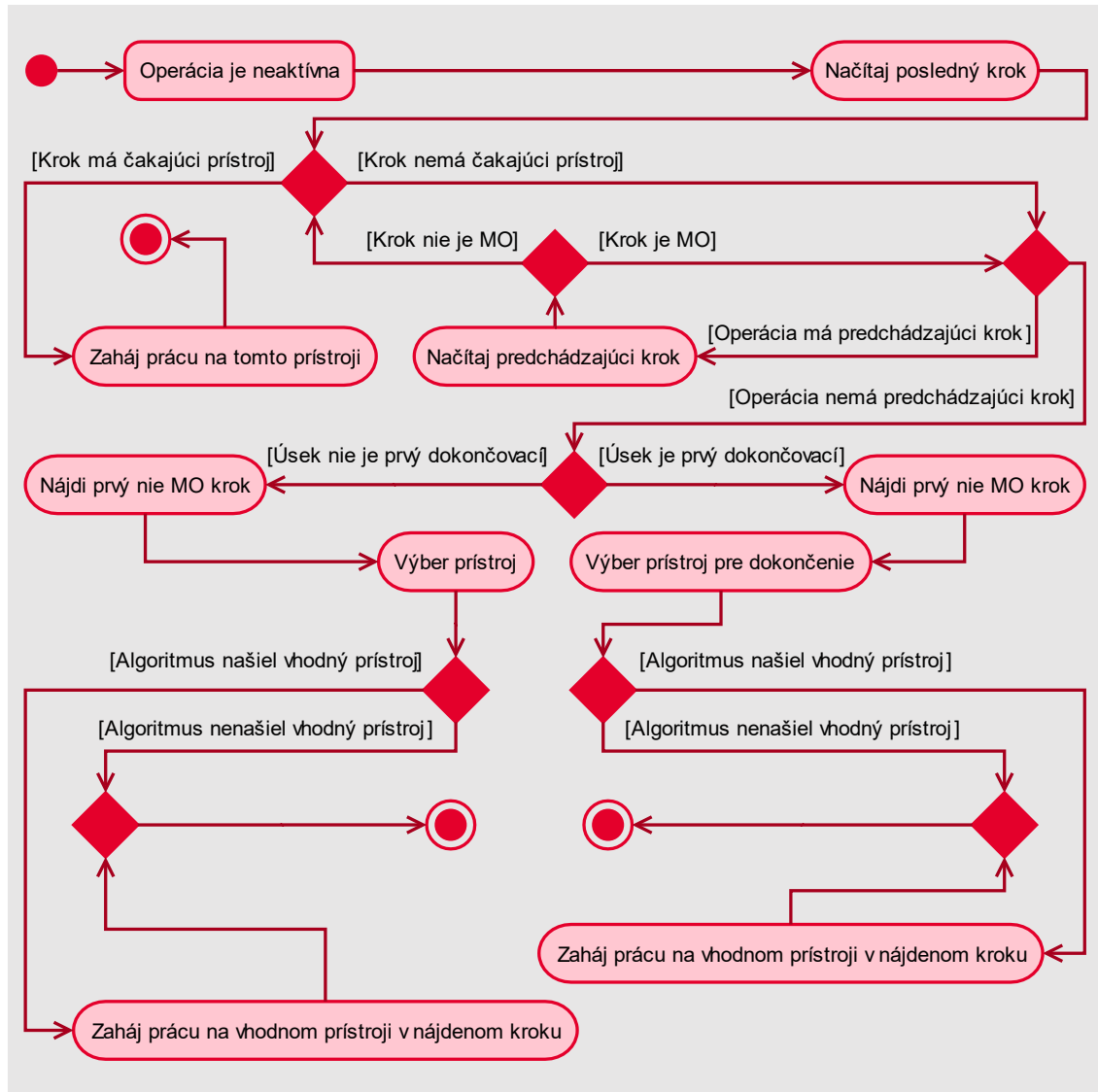
Pokiaľ je prístroj v prípravnej linke dochádza k volaniu metódy pre rozhodenie prístroja do opravných liniek. Metóda má názov *rozhodPrístrojDoLiniekOprava*. Algoritmy tejto metódy počítajú s rozdelením prístroja na viacero častí, do opravných liniek podľa percentuálnej pravdepodobnosti. Pre každú linku, do ktorej sa časť prístroja dostane sa o jedno navýši počítadlo kusov prístrojov v oprave tohto prístroja (premenná sa nachádza v štruktúre *PristrojVStredisku*, Obrázok 26). Po prechode má teda prístroj počítadlo nastavené na počet kusov, ktoré je potrebné opravnými linkami spracovať.

Ak sa prístroj nachádza v linke pre opravu prístroja, dochádza k volaniu metódy *prechodPristrojaOpravaDokoncenie*. Algoritmy tejto metódy otestujú, či je záznam o prístroji v zozname na dokončovacej linke. Pokiaľ sa záznam o prístroji už v dokončovacej linke nachádza, prístroj sa z tohto kroku len zmaže a počítadlo kusov prístrojov v oprave sa dekrementuje. Pokiaľ však záznam o prístroji na dokončovacej linke zatiaľ nie je, tak sa najskôr prístroj do tejto linky presunie a následne sa dekrementuje počítadlo.

Keď je prístroj na konci dokončovacej linky, je prístroj dokončený a opúšťa stredisko. Volá sa jednoduchá metóda *nastavPrístrojNaDokonceny*. Prístroj sa z daného kroku zmaže a do štruktúry *PristrojVStredisku* sa priradí číselný údaj o čase dokončenia a spolu s ním sa nastaví aj binárna hodnota stavu dokončenia.

### Priradzovanie prístrojov nevyťaženým technikom

Po presunutí dokončených prístrojov sa tieto prístroje zaradzujú na koniec fronty čakajúcich prístrojov na jednotlivé kroky a úseky. Následne sa obnovia stavy aktivity operácií. Technici, ktorí dokončili prácu na prístroji sú neaktívni a teda celá operácia na ktorej technik pracuje je neaktívna. Pre pokračovanie v chode strediska je potrebné zabezpečiť zahájenie práce na prístrojoch. Zjednodušený pohľad na aktivity vykonávané pre zahájenie práce na prístrojoch je na obrázku nižšie (Obrázok 28).



Obrázok 28 Zjednodušený diagram aktivít priradzovania prístrojov nevyťaženým technikom

V simulácií pri zahajovaní práce na prístrojoch nastávajú dva hlavné prípady. Vyplývajú z procesného modelu simulácie (*Dátová štruktúra simulácie*) a líšia sa pomyselnými krokmi, ktoré musí technik vykonať. Pokiaľ sa vhodné prístroje pre spracovanie nachádzajú v operácií, môže na nich technik priamo zahájiť prácu. Pokiaľ však technik spracoval všetky prístroje uložené vo frontách krokov svojej operácie, musí si prístroj priniesť z hlavnej fronty pre konkrétny úsek (konkrétnu časť linky). Samozrejme technik vyberá prístroj z tejto fronty, len pokiaľ prístroj dokáže spracovávať.



Špeciálny prípad pri braní prístrojov z hlavnej fronty pre konkrétny úsek nastáva pri fronte pred prvým úsekom v dokončovacej linke. Vo fronte sa nachádzajú všetky prístroje, ktoré vystúpia z opravných liniek strediska. Nemusia byť však kompletne. V tejto fronte sa môže nachádzať prístroj, ktorého nejaká časť je ešte stále v opravných linkách strediska. Z toho dôvodu musí technik najskôr zistiť, či má prístroj, ktorý je umiestnený vo fronte pred prvým úsekom dokončovacej linky všetky potrebné časti. Pokiaľ prístroju vo fronte pred týmto úsekom nejaké časti chýbajú, nemôže na ňom technik začať pracovať, až kým všetky chýbajúce časti neprídu. Musí si teda vybrať iný prístroj, ktorý má všetky časti a zároveň ktorý dokáže spracovať.

Pokiaľ technik v krokoch nemá žiaden čakajúci prístroj a taktiež pokiaľ nie je žiaden vhodný čakajúci prístroj ani na vstupnej fronte úseku, je technik neaktívny a teda operácia ostáva neaktívna. V takom prípade musí technik počkať, kým nepríde do fronty vhodný prístroj, ktorý by mohol začať spracovávať.

Diagram aktivít priradovania prístrojov (Obrázok 28) poskytuje podrobnejší a ucelený pohľad na jednotlivé aktivity, krok po kroku. V prvej aktivite je načítanie posledného kroku operácie. Prehľadávanie krokov od konca operácie je z dôvodu zabezpečenia rovnomerného a plynulého prechodu prístrojov operáciami. Pokiaľ by sa hľadal čakajúci prístroj od prvého kroku operácie, dochádzalo by na operáciách k hromadeniu čakajúcich prístrojov, v najhoršom prípade by mohlo na operácií dôjsť k nárastu fronty rovnako veľkej, ako samotnej vstupnej fronty prístrojov, čo by znemožňovalo prácu nadväzujúcim úsekom.

Pokiaľ má nejaký z krokov operácie čakajúci prístroj, spustí sa aktivita zahájenia práce na prístroji. Táto aktivita je jednoduchá, samotné algoritmy iba nastaví parameter *caka* na *false*, priradia tomuto prístroju číselný údaj o čase dokončenia z celkového času technika na operáciu upraveného percentuálnym pomerom dĺžky kroku, v iteráciách do premennej *casDoDokoncenia* a nastaví aktivitu príslušného kroku, operácie, časti linky a linky samotnej.

Pokiaľ však žiaden z krokov nemá čakajúci prístroj, pokračujú algoritmy hľadaním vhodného prístroja vo fronte úseku. Najskôr je potrebné zistiť, ktorým krokom začína technik v operácii pracovať. Pre tento účel slúži aktivita „Nájd prvý nie MO krok“, ktorej výstupom je číslo prvého technikom obsluhovaného kroku. Následne sa podľa toho, či je úsek bežný úsek strediska, alebo práve prvý úsek dokončovacej linky.

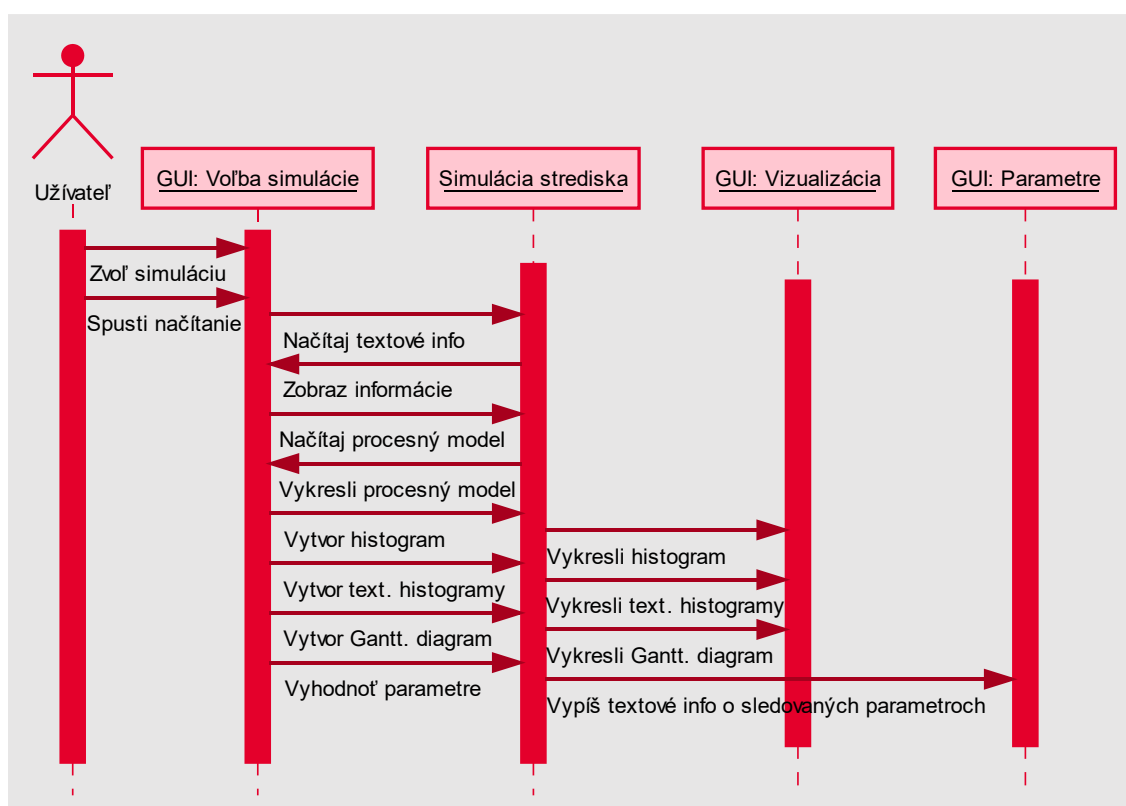
Ak je úsek bežný, vykonajú sa algoritmy obsluhujúce aktivitu *Vyber prístroj*. Algoritmy sú zabalené v metóde *VyberPrístroj* a jednoducho prechádzajú zoznamom prístrojov vo fronte podľa radenia FIFO a hľadajú prístroj, ktorý dokáže technik spracovávať. Pre zistenie schopnosti technika spracovať prístroj je opäť použitá metóda *canTechnikDoOpOn* z objektu databázy technikov. Pokiaľ algoritmy nájdu vhodný prístroj v tejto fronte, najskôr presunú prístroj do konkrétneho kroku a následne na ňom zahájajú prácu. Samotné zahájenie práce na prístroji prebieha podobne, ako vyššie popísané zahájenie práce na prístroji na kroku.

Ak je však úsek práve prvý úsek dokončovacej linky, vykonáva inú aktivitu pre výber prístroja. Táto aktivita obsahuje oproti aktivite pre obyčajný úsek navyše prehľadávanie zoznamu prístrojov v štruktúre *PrístrojVStredisku* (Obrázok 26) a ešte pred volaním *canTechnikDoOpOn* kontroluje, či má počítadlo *pocetKusovVOprave* nulovú hodnotu. Ostatné aktivity pracujú rovnako ako pri obyčajnom úseku.

### 3.2.9 Algoritmy pre vykresľovanie dát

Simulácia strediska poskytne základný prehľad o stavoch strediska pred začatím a po ukončení simulácie. Aplikácia taktiež umožňuje aj podrobnejšie skúmanie priebehu simulácie pomocou rôznych vizualizácií, počas ktorých sú užívateľovi poskytnuté zároveň aj textové informácie o sledovaných parametroch. Výhodou vizualizácie dát pomocou použitých algoritmov v aplikácii je rýchlosť vykreslenia.

Pre dosiahnutie vyššej rýchlosti vykresľovania algoritmy používajú údaje len v určitých krokoch simulácie. Nevýhodou je nižšie rozlíšenie, ktoré je závislé od veľkosti kroku a s tým spojená nižšia detailnosť vizualizácie. Pokiaľ by však užívateľ chcel podrobnejšie skúmať chod simulácie, aplikácia poskytuje možnosť exportu dát do štandardného csv formátu. Tieto dáta obsahujú údaje o každej jednej iterácii a môžu byť spracované pomocou ľubovoľného tabuľkového procesoru.



Obrázok 29 Zjednodušený sekvenčný diagram načítavania výsledkov simulácie

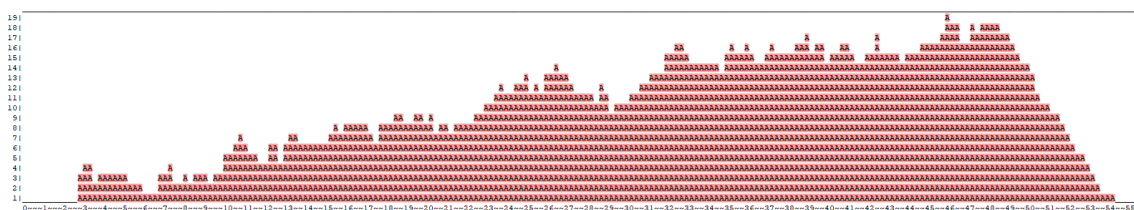
Obrázok vyššie (Obrázok 29) znázorňuje sekvenčný diagram priebehu algoritmu pre vizualizáciu priebehu dát z dátovej štruktúry simulácie. Vizualizačné algoritmy zahŕňa užívateľ v karte aplikácie pre voľbu načítania simulácie. Užívateľ zvolí niektorú z prebehnutých simulácií a túto voľbu potvrdí stlačením tlačidla pre načítanie výsledkov.

V prvom kroku dochádza k volaniu metódy *writeInfoToRTextCtrl*, ktorej algoritmy vypíšu súhrnné informácie o priebehu simulácie do textového elementu v danej karte. Následne je volaná metóda *writeProcesnyModelToGrid*, ktorá vykreslí procesný model do tabuľky taktiež v tejto karte. Ukážka takejto vizualizácie procesného modelu pomocou prvku tabuľky je ukázaná na obrázku tabuľky so skrytými popismi stĺpcov pre vykreslenie mapy procesného modelu (Obrázok 10).

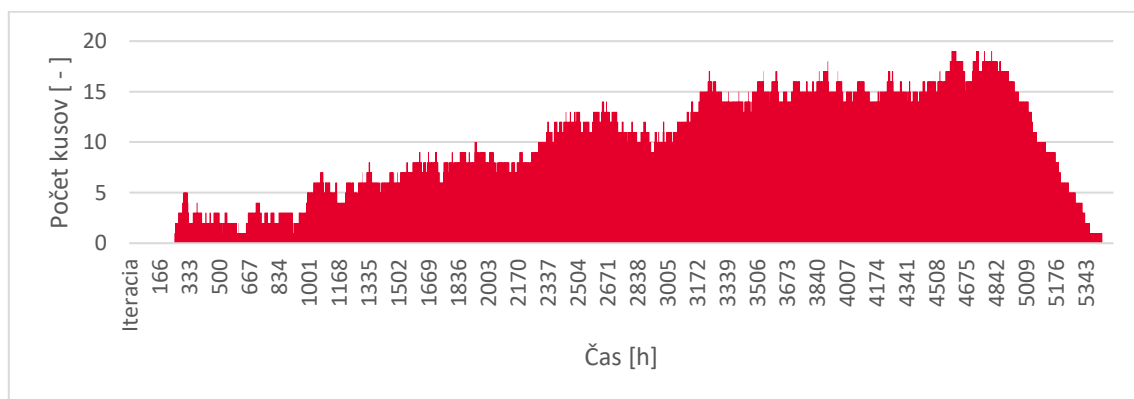
Ďalšie tri volané metódy vykresľujú priebeh do karty aplikácie s názvom *Vizualizácia chodu strediska v čase*. Prvá metóda ma názov *vytvorHist* a slúži na vytvorenie histogramu aktivity liniek v závislosti na čase simulácie do prvku tabuľky. Prechádza iterácie pre vybrané časy a počíta v nich počet aktívnych liniek. Pre každý vybraný čas vyfarbí v tabuľke na základe počtu aktívnych liniek príslušný počet riadkov.

Druhá metóda ma názov *vytvorTXTHist* a slúži na vykreslenie textových histogramov počtu prístrojov (Obrázok 30) na všetkých úsekoch strediska a príslušných histogramov vytťaženia technikov pre tieto úseky. Generovanie týchto histogramov je veľmi rýchle a textové prevedenie zároveň umožňuje užívateľovi tieto histogramy kopírovať a ukladať v textovom formáte. Ďalšou výhodou tohto prevedenia je možnosť umiestniť tieto histogramy do externého textového editoru pre možnosť prezerania týchto dát na celej obrazovke. Dáta sú rozsiahle, pretože obsahujú dva histogramy pre každý jeden úsek strediska.

Algoritmy vykresľovania textových grafov sa skladajú z dvoch hlavných častí. Prvá časť algoritmov prechádza dátovú štruktúru simulácie a vytvorí na základe údajov priamo na mieru rám grafu. Rám je zložený z orámovania spolu s číselným popisom osí. Druhá časť algoritmov vytvorí vnútro grafu. Taktiež prechádza dátovú štruktúru simulácie, zameriava sa však na iné parametre. Vnútro grafu je zložené z neviditeľných znakov tvoriacich medzery a viditeľných znakov reprezentujúcich údaje. Posledný riadok vnútra grafu je zložený zo znakov tvoriacich spodný okraj rámu a znakov tvoriacich posledný riadok. Pri vykresľovaní textových grafov sa tieto dve časti spoja a vytvoria jeden graf. K spájaniu častí dochádza k spoločnému vypisovaniu týchto častí po riadkoch. Najskôr sa vypíše riadok rámu, potom riadok vnútra grafu a až následne obdrží textový prvok znak pre nový riadok. Vypisovanie riadku vnútra je posunuté o počet riadkov histogramu, v ktorých sa nachádza len rám, bez vnútra.



Obrázok 30 Zmenšená ukážka textového histogramu v externom textovom editore



Graf 1 Histogram vytvorený z exportovaného csv pomocou externého tabuľkového procesoru

Na obrázku a grafe vyššie (Obrázok 30 a Graf 1) sú zobrazené možnosti vizualizácie histogramov. Dáta použité v ukážkach sú rovnaké. Porovnaním je možné vidieť, že graf vygenerovaný tabuľkovým procesorom z csv súboru obsahuje viac detailov, ako textový histogram vygenerovaný aplikáciou, avšak textový histogram má správny tvar a pre rýchlu vizualizáciu je dostačujúci. Pre porovnávanie chodov jednotlivých scenárov simulácií prevažuje potreba rýchleho vykresľovania nad potrebou vyšších detailov.

Tretia metóda slúži na vykreslenie Ganttovho diagramu do prvku tabuľky. Metóda má názov *spravGanttGrid* a ukážka jej výstupu je v práci na obrázku Ukážka využitia tabuľky pre vykreslenie Ganttovho diagramu (Obrázok 11). Jej algoritmy prechádzajú štruktúru vo vybraných iteráciách a rôznymi farbami vyfarbujú bunky riadkov tabuľky. Farby sú v Ganttovom diagrame použité štyri. Zelená, modrá, sivá, červená a slúžia na vyjadrenie činnosti liniek a úsekov, vyjadrenie nečinnosti a farebné oddelenie liniek.

Posledná volaná metóda slúži na vypočítanie a vypísanie textových informácií o sledovaných parametroch do karty užívateľského rozhrania s názvom *Sledované parametre*. Metóda má názov *vypisPristrojeDokonceneNedokoncene*. Algoritmy prechádzajú dátovú štruktúru simulácie a postupne vypisujú prístroje s informáciami.

Zoznamom prístrojov algoritmy prechádzajú v dvoch slučkách. V prvej slučke dochádza k vypisovaniu dokončených prístrojov. Pri každom dokončenom prístroji sa taktiež vypíše aj údaj o prechode. Pokiaľ bol prístroj pred zahájením simulácie umiestnený na vstupe do strediska, vypíše sa údaj o dĺžke prechodu prístroja strediskom, pokiaľ bol prístroj pred začatím simulácie umiestnený na inom úseku, vypíše sa informácia, že sa jedná o prístroj, ktorý bol pred začiatkom simulácie rozpracovaný. Po vykonaní algoritmov pre prvé prechádzanie prístrojov sa pod zoznam prístrojov vypíše počet dokončených prístrojov.

Druhá slučka pri prechádzaní prístrojov vypisuje prístroje, ktoré sú aj po uplynutí doby simulácie stále prítomné v stredisku. Prístroje, ktoré sa po skončení simulácie aj naďalej nachádzajú v stredisku sú WIP. Keď sa dokončí vypisovanie, tak sa pod vypísaný zoznam napíše informácia o počte rozpracovaných prístrojov.

## 4 TESTOVANIE APLIKÁCIE, SKÚŠANIE A OVEROVANIE

Aplikácia bola rôznymi okrajovými stavmi testovaná už počas samotného vývoju. Takéto testovanie sa používa pre vyhľadávanie na prvý pohľad prehliadateľných chýb algoritmov spôsobených neošetrenými stavmi. Tieto chyby sú ľahko odhaliteľné, pretože keď sa vyskytnú, v aplikácii nastane očividný problém napríklad s chybným prístupom k adresám v pamäti pre štruktúru, kedy sa namiesto údajov o názve prístroja objaví v štruktúre meno technika, alebo iný nezmyselný údaj (táto chyba nastáva pri mazaní údajov v štruktúre a následnom čítaní neplatných adries pamäte). Pri iných závažných chybách dôjde k pádom aplikácia a podobne.

Okrem takéhoto typu testovania je však nutné aplikáciu otestovať aj na správnosť chodu. Testovanie tohto typu sa uskutočňuje zvyčajne po dosiahnutí určitého štádia vývoja aplikácie, prípadne jej časti. Vyhľadáva funkčné chyby, ktoré síce nespôsobujú viditeľné problémy, avšak majú vplyv napríklad na zavedenie chyby do výsledku. Funkčné testovanie býva zložitejšie na nájdenie chýb, pretože pri ňom nenastane žiadna očividná chyba. Príkladom chyby, ktorá bola odhalená funkčným testovaním v neskoršom štádiu vývoja bolo nesprávne presúvanie prístrojov počas simulácie z linky oprava. Táto chyba sa vyznačovala tým, že prístroj po prechode opravnou linkou namiesto, aby sa zaradil do zoznamu čakajúcich na dokončenie, bol presunutý do ďalšej opravnej linky v stredisku. Na prvý pohľad nebolo možné pozorovať chybu v aplikácii, pretože simulácia prebehla v poriadku, bez akýchkoľvek problémov. Výpis dát sa taktiež zdal byť v poriadku. Chyba bola odhalená pri podrobnejšom testovaní samotných krokov algoritmu. Pri krokovanií simulácie po jednotlivých iteráciách bolo zistené, že prístroj po dokončení posledného kroku opravnej linky zmizne bez toho, aby sa objavil v zozname prístrojov na vstupe dokončovacej linky. Následne bol objavený na začiatku ďalšej opravnej linky. Pre podrobnejšie preskúmanie tohto problému bola simulácia ďalej kroková. Po prechode touto opravnou linkou sa objavil znova na ďalšej opravnej linke. Nakoniec až po prechode poslednou opravnou linkou sa pridá do zoznamu čakajúcich prístrojov na dokončovacej linke. Toto správanie napomohlo vo vytvorení predstavy o chybe, jej následnej lokalizácii a oprave.

Testovanie aplikácie na chyby je však časovo veľmi náročná úloha, ktorej sa v praxi venujú celé samostatné tímy pracovníkov, ako už bolo spomenuté v podkapitole 3.2.2 Testovanie a ošetrovanie stavov. Táto kapitola sa zaoberá otestovaním poslednej verzie aplikácie na základnú funkčnosť samotných algoritmov a následným overovaním funkčnosti algoritmov. Prvá podkapitola rozoberá testovanie základnej funkčnosti jednotlivých častí aplikácie spolu s popisom požadovaných sledovaných vlastností týchto častí. Druhá podkapitola sa venuje overeniu samotných výsledkov a funkcionality aplikácie ako celku, z kvantitatívneho hľadiska.

## 4.1 Testovanie funkčnosti algoritmov

Základné testovanie funkčnosti vychádza z troch modelov, ktoré sa nachádzajú v prílohe. Tieto súbory boli vytvorené pre porovnanie algoritmov na odhad času s výsledkom simulácie v podkapitole *Porovnanie odhadov s reálnym priebehom*.

### 4.1.1 Prirad'ovacie algoritmy

Zmazaním všetkých technikov z operácií dôjde k vytvoreniu prázdneho strediska. Model bol uložený pod názvom „model\_bez\_tech\_250\_pristrojov\_na\_vstupe.modp“ a pridaný do prílohy. Na obrázku nižšie (Obrázok 31) je zobrazený zmenšený model strediska bez technikov. Obrázok ukazuje všetky úseky vyfarbené červenou farbou, pretože nedokážu spracovať všetky požadované prístroje.

Vstupna kontrola	Diagnostika	Rozobratie	Vymena tesneni	Mazanie	Zlozenie	Opt+El kontrola	Funkcna kontrola	Vystupna kontrola
Diagnostika dosky	Vybratie dosky	Vymena dosky	Testovanie dosky					
Odstranenie konektoru	Vymena konektoru	Kontrola konektoru						

Obrázok 31 Zmenšený náhľad na stredisko bez technikov

Prirad'ovací algoritmus najskôr vytvorí platný model a následne zaháji jednoduché optimalizované prirad'ovanie. Tieto algoritmy sú podrobne vysvetlené v podkapitole 3.2.6 Automatizované prirad'ovanie technikov (Obrázok 21). Automatizovaným priradením desiatich technikov do platného modelu bude rozloženie nasledovné:

--- Linka 0 ---			
Vstupna kontrola	Vaughan Whitaker	Zachery Ferguson	
Diagnostika	Hall Skinner		
Rozobratie	Jeremy Farmer	Ivor Britt	
--- Linka 1 ---			
Vymena tesneni	Hedy Knight	Mollie Hodge	
Mazanie	Chase Harding	Hector Savage	
--- Linka 2 ---			
Diagnostika dosky	Anne Bentley		
Vybratie dosky	Meredith Mckenzie		
Vymena dosky	Dora McClure		
Testovanie dosky	MacKenzie Stephens		
--- Linka 3 ---			
Odstranenie konektoru	Adria Green		
Vymena konektoru	Abra Yang		
Kontrola konektoru	Chandler Waller		
--- Linka 4 ---			
Zlozenie	Leigh Cooke	Walter Swanson	
Opt+El kontrola	Jack Carpenter	Natalie Rutledge	Kathleen Bray
Funkcna kontrola	Ilina Allen	Nina Aguilar	
Vystupna kontrola	Jaquelyn Kennedy	Gareth Stout	Constance Parsons

Obrázok 32 Automatizované priradenie desiatich technikov na úseky

Na obrázku vyššie (Obrázok 32) je možné vidieť, že desiatich technici boli rozložený na úseky optimalizovane. Úseky s väčším intervalom boli pri priradovaní technikov uprednostnené. Následným ďalším priradením všetkých technikov vznikne hrubo rozložený optimalizovaný zoznam technikov. Jednotlivé riadky sú pre lepšiu orientáciu farebné oddelené. Zmenšený náhľad tohto zoznamu je na obrázku nižšie:

--- Linka 0 ---													
Vstupná kontrola	Vaughan Whitaker	Zachery Ferguson	Iona Lopez	Yael Montoya	Amethyst Sloan	Ethan Eaton	Karly Mills	MacKenzie Howard	Angela Sweet				
Diagnostika	Hall Skinner	Timon Whitney	Troy Cotton										
Rozobratie	Jeremy Farmer	Ivor Britt	Kyra Daniels	Macy Nguyen	Orlando Myers	Carl Phillips	Forrest Bailey	Jana Obrien	Preston Morris				
--- Linka 1 ---													
Vymena tesneni	Hedy Knight	Mollie Hodge	Kirsten Santiago	Cole Edwards	Grady Fox	Bianna Robinson	Nevada Arnold	Illiana Fleming	Tiger Vance	Catherine Cook	Hyacinth Herrera	Christian Knight	
Mazanie	Chase Harding	Hector Savage	Porter McBride	Yoshi Mason	Keith Barnes	Ronan Buck	Lucy Chandler	Brenden Mathews	Nora Rosa	Brooke Frazier	Aileen Bryan	Ira Harrell	
--- Linka 2 ---													
Diagnostika dosky	Anne Bentley	Jarrod Miranda	Mufutau Cervantes										
Vybratie dosky	Meredith McKenzie	Declan Rose	Sawyer Duffy	Silas Haney	Jason Gregory								
Vymena dosky	Dora McClure	Deanna Davis											
Testovanie dosky	MacKenzie Stephens	Nasim Guerrero	Todd Rogers	Barry Marsh	Bradley Decker								
--- Linka 3 ---													
Odstranenie konektoru	Adria Green												
Vymena konektoru	Abra Yang												
Kontrola konektoru	Chandler Waller												
--- Linka 4 ---													
Zoznenie	Leigh Cooke	Walter Swanson	Micha Beasley	Harper Watkins	Leroy Sweet	Logan Soto	Suki Bradford						
Opt-El kontrola	Jack Carpenter	Natalie Rutledge	Kathleen Bray	Katell Washington	Imogene Burnett	Beverly Wolf	Julian Phillips	Emmanuel Lucas	Ann Patel	Hollee Emerson	Savannah Greer		
Funkcna kontrola	Iiana Allen	Nina Aguilar	Ingrid Price	Unity Garner	Lynn Malone	Graham Jimenez	Martina Hatfield						
Vystupna kontrola	Jaquelyn Kennedy	Gareth Stout	Constance Parsons	Jarrod Wilkinson	Jonah Talley	Jillian Ball	Maia Finch	Mira Valenzuela	Shaeleigh Justice	Gil Oconnor	Jared Odonnell	Cynthia Buckner	

Obrázok 33 Zmenšený náhľad zoznamu po automatizovanom priradení všetkých technikov

Zoznam je pomerne veľký a preto je jeho obrázok (Obrázok 33) značne zmenšený. Je na ňom však vidieť predelenie liniek – červené horizontálne čiary, podľa ktorých je možné rozoznať jednotlivé linky. Optimalizované rozloženie je možné vidieť napríklad v prípade linky číslo 3, ktorá má priradeného len jedného technika na každý úsek. Linka totižto obsahuje veľmi krátke operácie a teda nie je potrebné, aby na úsekoch boli ďalší technici (Obrázok 37). Podobne môžeme predpokladať, že linka číslo 1 má vzhľadom na frontu prístrojov priemerne najdlhšie operácie, keďže má najviac priradených technikov. Posledný úsek na poslednej linke má rovnaký počet technikov a teda vzhľadom na celkový veľký počet dobre zaškolených technikov v stredisku môžeme predpokladať, že bude mať úsek podobne dlhý čas. Problematika časov prechodov jednotlivých prístrojov z fronty je zobrazená v podkapitole 4.1.3 Simulácia prechodu jedného prístroja strediskom. V podkapitole sa taktiež nachádza obrázok zobrazujúci konkrétnu ukážku dĺžok časov pre dva vybrané prístroje (Obrázok 37).

Tento test ukázal, že priradovacie algoritmy fungujú. Automatizovane priradujú technikov do strediska. Rovnako, ako dokážu priradiť len pár technikov, tak dokážu priradiť aj všetkých technikov. Taktiež je viditeľné, že nerozmiestňujú technikov rovnomerne, ale s určitým vzorcom. Podľa analýzy tohto vzorca a overení predpokladov z časti simulácie pre zobrazenie prechodu jedného prístroja, môžeme usúdiť, že vzorcami vytvorené vzory majú viditeľnú závislosť distribúcie v závislosti na dĺžke operácie vykonávanej na konkrétnom úseku. Takto docielia optimalizované rozloženie pracovnej sily naprieč operáciami na úsekoch celého strediska. Vyhľadujú rozdiely dĺžok intervalov a nastoľujú rovnováhu medzi úsekmi pre umožnenie vzniku rýchlejšieho a hladšieho prechodu. Táto rovnováha by však mala predstavovať len hrubé rozloženie, ktoré môže byť ďalej jemnejšie optimalizované.



Tieto algoritmy vykonajú dodatočnú optimalizáciu pre rýchlosť prechodu. Ponúkajú hrubú a jemnú optimalizáciu. Hrubá optimalizácia premiestňuje technikov medzi úsekmi a teda mení aj počty technikov na úsekoch. Jemná optimalizácia vymieňa technikov medzi úsekmi. Počet technikov sa teda jemnou optimalizáciou nemení, ale poprehadzuje sa ich umiestnenie.

Obrázok 34 Zmenšený náhľad na zoznam technikov po vykonaní optimalizácie

Pred optimalizáciou	Po optimalizácií
Dora Mcclure	Dora Mcclure
Deanna Davis	Deanna Davis
	Tiger Vance

Na treťom úseku linky číslo 2 (úsek pre výmenu dosky) došlo len k presunu technika. Na tomto úseku sa teda vykonala len hrubá optimalizácia. Jemná optimalizácia pre tento úsek nebola potrebná. V príslušnej tabuľke (Tabuľka 7) je teda možné vidieť po vykonaní optimalizácie len pribudnutie jedného technika.

Tabuľka 8 Technici na úsekoch v linke 3



V linke číslo 3 je možné vidieť len jedného priradeného technika na úsekoch. Úseky na tejto linke sú krátke, na výkonnosť strediska nemajú priamy vplyv, teda nie je potrebné technikov pridávať a ani meniť. Po optimalizácii je však možné vidieť, že sú na tento úsek priradení úplne iní technici (Tabuľka 8). Výmenu technikov mali za následok algoritmy jemnej optimalizácie, ktoré vypočítali, že by bolo vhodnejšie mať týchto technikov na inom úseku. Nejedná sa teda o optimalizáciu týchto úsekov, ale o využitie týchto technikov pri optimalizácii iných úsekov strediska.

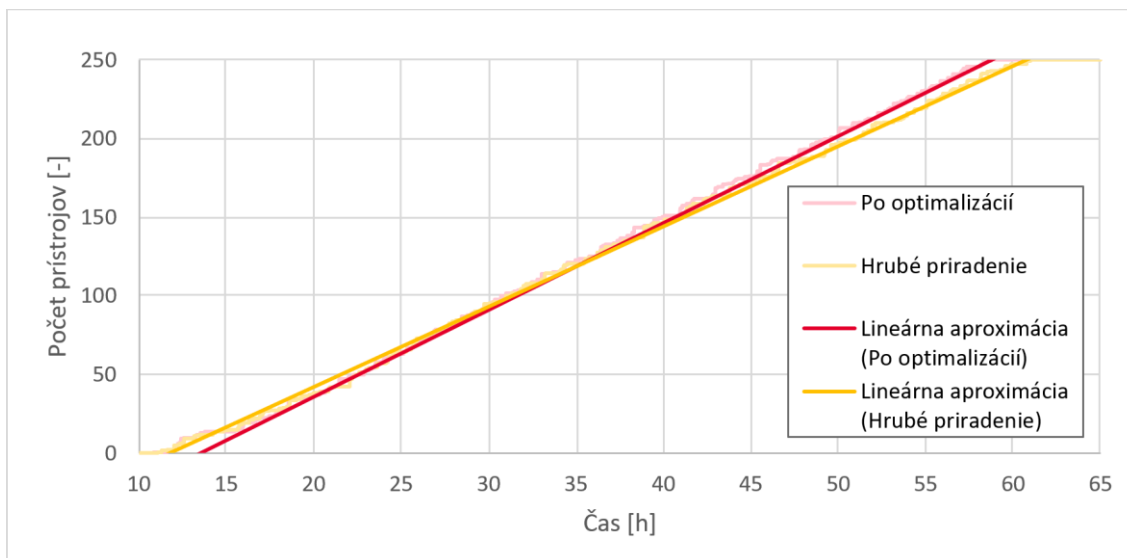
Pred optimalizáciou	Po optimalizácii
Leigh Cooke	Leroy Sweet
Walter Swanson	Suki Bradford
Micah Beasley	Jillian Ball
Harper Watkins	Mira Valenzuela
Leroy Sweet	Walter Swanson
Logan Soto	Jarrod Wilkinson
Suki Bradford	Lynn Malone

Tabuľka 9 Technici na úseku zloženie

Prvý úsek dokončovacej linky určený pre skladanie prístrojov má po optimalizácii taktiež zamenených technikov (Tabuľka 9). V tomto prípade však išlo o zmenu technikov z dôvodu optimalizácie daného úseku. Pred optimalizáciou bol interval strediska 0,189 a interval tohto úseku 0,172. Úseku bolo teda potrebné znížiť výkonnosť. Koeficient odobratia technika mal tento úsek 0,029, čo znamená, že odobratím technika by úsek nadobudol interval 0,201, omnoho väčší, ako celkový interval strediska. Z toho dôvodu nie je možné použiť hrubú optimalizáciu a na úseku sa prejavili zmeny len jemnej optimalizácie. Po vykonaní optimalizačných algoritmov je interval strediska presnejšie na 4 desatinné miesta 0,1801 a interval úseku 0,1800. Jemná optimalizácia zámenou technikov teda nastavila interval úseku celkom presne k intervalu strediska.

Optimalizácia bola pri tomto testovaní nastavená na 30 iterácií. S počtom optimalizácií sa dá experimentovať, užívateľ si môže vyskúšať, aký počet iterácií optimalizačného algoritmu bude najviac vyhovovať jeho modelu strediska, počtu technikov v stredisku a prístrojom v stredisku. Dĺžka priebehu tohto algoritmu je rôzna. Závisí od zložitosti celého systému, ktorý musí optimalizovať, od počtu prístrojov, technikov a od zložitosti modelu.

Pre detailnejšie otestovanie funkčnosti optimalizačných algoritmov bola následne vykonaná samotná simulácia oboch rozložení technikov. Pred optimalizáciou rozloženia technikov bola doba prechodu všetkých prístrojov strediskom 61 hodín a 0 minút. Po optimalizácii na rýchlosť priebehu bol simuláciou zistený čas prechodu všetkých prístrojov 58 hodín a 47 minút. Optimalizačné algoritmy teda presunutím a zámenou technikov dosiahli rýchlejšie spracovanie fronty prístrojov o viac ako dve hodiny.



Graf 2 Prechod prístrojov strediskom

Grafická ilustrácia priebehu simulácie je uvedená v grafe prechodu prístrojov strediskom (Graf 2). Údaje pre vykreslenie grafu boli získané z dátových súborov priebehu exportovaných aplikáciou do csv formátu. V tomto grafe sú zobrazené lineárne aproximácie kumulatívneho počtu dokončených prístrojov pre každú vzorku dát. Optimalizovaný priebeh má oproti priebehu po počiatočnom hrubom priradení strmšiu smernicu, čo znamená rýchlejšie spracovanie fronty prístrojov. Rýchlejšie spracovanie fronty značí efektívnejšie rozloženie technikov, vyššiu priepustnosť strediska a teda funkčnosť optimalizačných algoritmov je potvrdená.

### 4.1.3 Simulácia prechodu jedného prístroja strediskom

Tento typ simulácie poskytuje jednoduché zobrazenie strediska, pri ktorom je na každom úseku priradený jeden technik (prvý technik zo zoznamu technikov na úseku) a skúma prechod jedného prístroja. Každému technikovi je priradený čas, ako na mape modelu, tak aj v textovej forme spolu s výpočtami časov celých liniek a celkovým časom prechodom strediskom. V mape modelu je taktiež zvýraznená aj kritická cesta strediskom. Kritická cesta je zvýraznená červenou farbou, ukážka zvýraznenia kritickej cesty pre prístroj *Anie-8-ed5709* je na zmenšenom obrázku nižšie (Obrázok 35).

Výsledky simulácie odozvy strediska na prístroj. Kritická cesta je zvýraznená červenou linkou:

Vstupná kontrola	Diagnostika	Rozobratie	Vymena tesneni	Mazanie	Zlozenie	Opt+El kontrola	Funkčna kontrola	Výstupná kontrola
1h 43m 48s.	0h 31m 48s.	1h 38m 24s.	1h 46m 48s.	2h 22m 48s.	1h 7m 48s.	2h 17m 60s.	1h 6m 0s.	2h 28m 12s.
	Diagnostika dosky	Vybratie dosky	Vymena dosky	Testovanie dosky				
	0h 25m 12s.	1h 13m 12s.	0h 35m 24s.	1h 8m 60s.				
	Odstranenie konektoru	Vymena konektoru	Kontrola konektoru					
	0h 31m 48s.	0h 12m 36s.	0h 33m 36s.					

Obrázok 35 Zmenšený náhľad simulácie prechodu jedného prístroja Anie-8-ed5709 strediskom

Kritická cesta sa môže v závislosti na konkrétnom prístroji meniť. Každý prístroj totižto môže mať rôzne dlhé trvania operácií na úsekoch, čo môže rôzne rozhodnúť celkové dĺžky prechodu prístroja linkami. Pre prípad tohto modelu strediska je zmena kritickej cesty viditeľná pri prístroji *Fixy-6-6i3817* (Obrázok 36). Kritická cesta pre tento prístroj nejde linkou číslo 1, ako tomu bolo pre prístroj *Anie-8-ed5709*, ale linkou číslo 2.

Výsledky simulácie odozvy strediska na prístroj. Kritická cesta je zvýraznená červenou linkou:

Vstupná kontrola	Diagnostika	Rozobratie	Vymena tesneni	Mazanie	Zlozenie	Opt+El kontrola	Funkcna kontrola	Vystupna kontrola
2h 4m 12s.	0h 47m 24s.	2h 4m 12s.	2h 7m 48s.	2h 47m 24s.	1h 37m 12s.	2h 30m 36s.	1h 46m 12s.	2h 31m 12s.
	Diagnostika dosky	Vybratie dosky	Vymena dosky	Testovanie dosky				
	0h 43m 12s.	1h 47m 24s.	0h 52m 12s.	1h 46m 48s.				
	Odstranenie konektoru	Vymena konektoru	Kontrola konektoru					
	0h 47m 24s.	0h 25m 12s.	0h 48m 36s.					

Obrázok 36 Zmenšený náhľad simulácie prechodu jedného prístroja *Fixy-6-6i3817* strediskom

Celkový prehľad všetkých časov v prvku textového zoznamu je pre porovnanie oboch prístrojov uvedený na obrázku nižšie (Obrázok 37). Tieto časy korešpondujú s časmi zo zmenšených obrázkov (Obrázok 35 a Obrázok 36). Pre prístroj *Anie-8-ed5709* je možné odčítať najdlhší čas linky (linka číslo 1) v opravnej časti strediska, presne 4 hodiny a 9 minút. Prístroj *Fixy-6-6i3817* má najdlhšiu opravnú linku (linku číslo 2) s časom 5 hodín a 9 minút. Ostatné opravné linky sú kratšie, ležia mimo kritickej cesty a teda nie sú podfarbené červenou farbou. Prípravná a dokončovacia linka je zafarbená vždy. Tieto linky nemajú žiadnu paralelnú linku a preto sú tieto linky vždy súčasťou kritickej cesty.

Textove informacie:	Textove informacie:
Cas prechodu pristroja strediskom je 15h 3m 36s.	Cas prechodu pristroja strediskom je 18h 30m 36s.
Cas prechodu pristroja linkou cislo 0 je 3h 53m 60s.	Cas prechodu pristroja linkou cislo 0 je 4h 55m 48s.
Cas prechodu pristroja linkou cislo 1 je 4h 9m 36s.	Cas prechodu pristroja linkou cislo 1 je 4h 55m 12s.
Cas prechodu pristroja linkou cislo 2 je 3h 22m 48s.	Cas prechodu pristroja linkou cislo 2 je 5h 9m 36s.
Cas prechodu pristroja linkou cislo 3 je 1h 18m 0s.	Cas prechodu pristroja linkou cislo 3 je 2h 1m 12s.
Cas prechodu pristroja linkou cislo 4 je 7h 0m 0s.	Cas prechodu pristroja linkou cislo 4 je 8h 25m 12s.
Operacia Vstupna kontrola trva 1h 43m 48s.	Operacia Vstupna kontrola trva 2h 4m 12s.
Operacia Diagnostika trva 0h 31m 48s.	Operacia Diagnostika trva 0h 47m 24s.
Operacia Rozobratie trva 1h 38m 24s.	Operacia Rozobratie trva 2h 4m 12s.
Operacia Vymena tesneni trva 1h 46m 48s.	Operacia Vymena tesneni trva 2h 7m 48s.
Operacia Mazanie trva 2h 22m 48s.	Operacia Mazanie trva 2h 47m 24s.
Operacia Diagnostika dosky trva 0h 25m 12s.	Operacia Diagnostika dosky trva 0h 43m 12s.
Operacia Vybratie dosky trva 1h 13m 12s.	Operacia Vybratie dosky trva 1h 47m 24s.
Operacia Vymena dosky trva 0h 35m 24s.	Operacia Vymena dosky trva 0h 52m 12s.
Operacia Testovanie dosky trva 1h 8m 60s.	Operacia Testovanie dosky trva 1h 46m 48s.
Operacia Odstranenie konektoru trva 0h 31m 48s.	Operacia Odstranenie konektoru trva 0h 47m 24s.
Operacia Vymena konektoru trva 0h 12m 36s.	Operacia Vymena konektoru trva 0h 25m 12s.
Operacia Kontrola konektoru trva 0h 33m 36s.	Operacia Kontrola konektoru trva 0h 48m 36s.
Operacia Zlozenie trva 1h 7m 48s.	Operacia Zlozenie trva 1h 37m 12s.
Operacia Opt+El kontrola trva 2h 17m 60s.	Operacia Opt+El kontrola trva 2h 30m 36s.
Operacia Funkcna kontrola trva 1h 6m 0s.	Operacia Funkcna kontrola trva 1h 46m 12s.
Operacia Vystupna kontrola trva 2h 28m 12s.	Operacia Vystupna kontrola trva 2h 31m 12s.

Obrázok 37 Porovnanie časov z GUI aplikácie pre prístroj *Anie-8-ed5709* a *Fixy-6-6i3817*

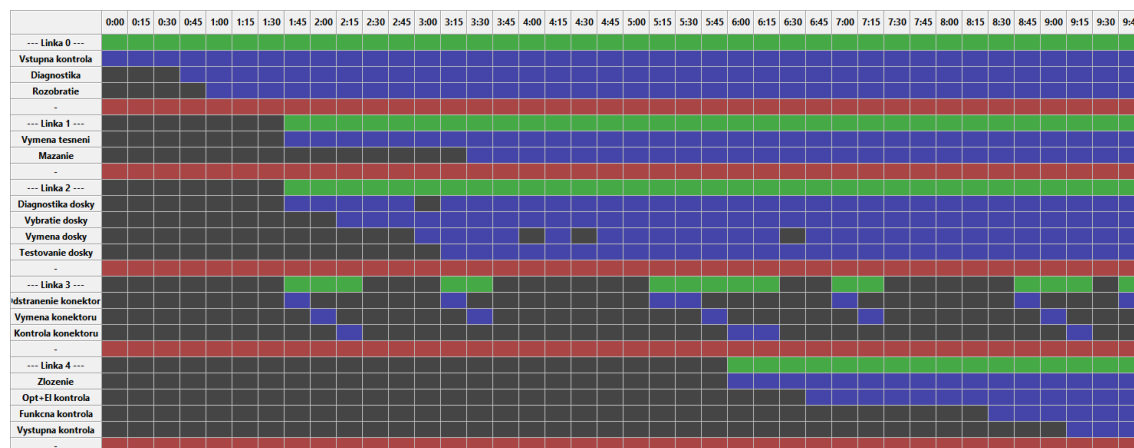
V textových informáciách je možné taktiež vidieť údaj o celkovom čase prechodu prístroja strediskom. Jedná sa o dĺžku kritickej cesty. Výpočet je zložený zo súčtu dĺžok prechodu prípravnou, opravnou a najdlhšou linkou v časti oprava.

#### 4.1.4 Simulácia chodu strediska

V tejto simulácii sa vykonáva postupné krokované prístrojov strediskom. Podľa nastavených parametrov a dĺžky simulácie je možné vykonať simuláciu prechodu prístrojov strediskom. Podľa dĺžky simulácie je možné nastaviť simulovanie jedného pracovného dňa, alebo pri vhodných parametroch a dlhšom čase simulácie taktiež aj prechod celej fronty prístrojov.

Pre testovanie funkčnosti simulácie v tejto kapitole bolo nastavené rozloženie prístrojov len na vstup strediska. Všetky prístroje sa teda pred zahájením simulácie nachádzali vo vstupnej fronte strediska. Pri takomto nastavení je totižto možné lepšie sledovať prácu algoritmov. Keďže sa všetky prístroje nachádzajú výhradne len na vstupe, je možné sledovať postupné zaplňovanie strediska a nábehy liniek. Čas simulácie bol pre túto kapitolu taktiež nastavený na prechod všetkých prístrojov strediskom, aby bolo možné sledovať postupné vypínanie liniek.

Na začiatku sa dá pozorovať postupné naplňovanie strediska prístrojmi. Naplňovanie strediska je možné vidieť počas prvých pár hodín Ganttovho diagramu. Zmenšený obrázok začiatku Ganttovho diagramu je zobrazený na obrázku nižšie (Obrázok 38). Je na ňom možné vidieť postupné spúšťanie celých liniek v rôznych časoch (zelené línie) a taktiež samotnú aktivitu práce na jednotlivých úsekoch (modré políčka).



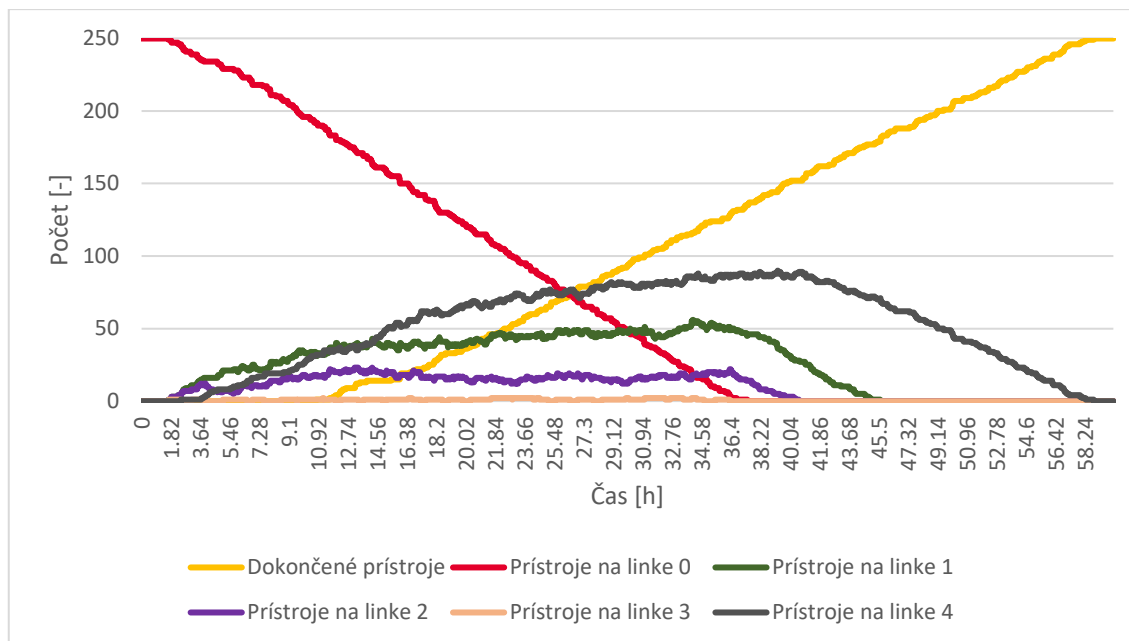
Obrázok 38 Zmenšený náhľad Ganttovho diagramu postupného spúšťania liniek a úsekov

Sivé políčka, ktoré sa občas nachádzajú v úsekoch značia prestoje úsekov. V sivom políčku je úsek neaktívny, čo znamená, že technici aktuálne nepracujú a čakajú na prístroj. Čím menej sivých políčok Ganttov diagram obsahuje, tým je chod strediska efektívnejší. V extrémnych prípadoch môže dôjsť k vytvoreniu celých okien prestojov zložených zo sivých políčok.

V jednej linke na obrázku (Obrázok 38) je možné vidieť výrazne vzory a jednotlivé úseky sú, podobne ako celá linka, väčšinu času neaktívne. Táto skutočnosť je spôsobená výrazným nepomerom výkonu linky k ostatným linkám. Je to linka číslo 3, ktorá bola v tejto kapitole už veľa krát spomínaná. Má najkratšie operácie na úsekoch (Obrázok 37) a každý úsek obsluhuje iba jeden technik (Tabuľka 8). Tento model je vlastný ukážkový model strediska pre demonštrovanie aplikácie, pokiaľ by sa však takýto vzor vyskytol v reálnom modeli strediska, bolo by vhodné prehodnotiť chod tejto linky, jej spúšťanie len v určitých intervaloch, prípadne zlúčenie úsekov pre obsluhu jedným technikom.

### Analýza prechodu prístrojov

Výsledky simulácie je možné podrobnejšie analyzovať pomocou vykreslenia detailných grafov v externom tabuľkovom procesore. Pre všetky ukážky grafov v tejto sekcii sú použité dáta výhradne len z exportovaného csv súboru.



Graf 3 Celkový pohľad na prechod všetkých prístrojov strediskom

Prvým typom grafu, ktorý je možné vykresliť je prehľad počtov prístrojov v stredisku (Graf 3). Takýto graf sa dá využiť na rýchly prehľad o správaní sa strediska v najvyššej vrstve – samotných liniek. Čím vyššieho bodu krivka dosiahne, tým bolo na danom úseku väčšie hromadenie. V grafe sú dve krivky obsahujúce najvyšší bod pri hodnote 250 prístrojov. Táto hodnota je celkový počet prístrojov, červená krivka ho zo začiatku obsahuje, pretože sa na nej vyskytuje celá vstupná fronta. Žltá krivka zas tento počet obsahuje, pretože reprezentuje kumulatívny celkový počet prístrojov, ktoré boli dokončené. Keďže simulácia bola nastavená na čas pre prechod všetkých prístrojov, táto krivka na konci dosahuje hodnoty počtu všetkých prístrojov.

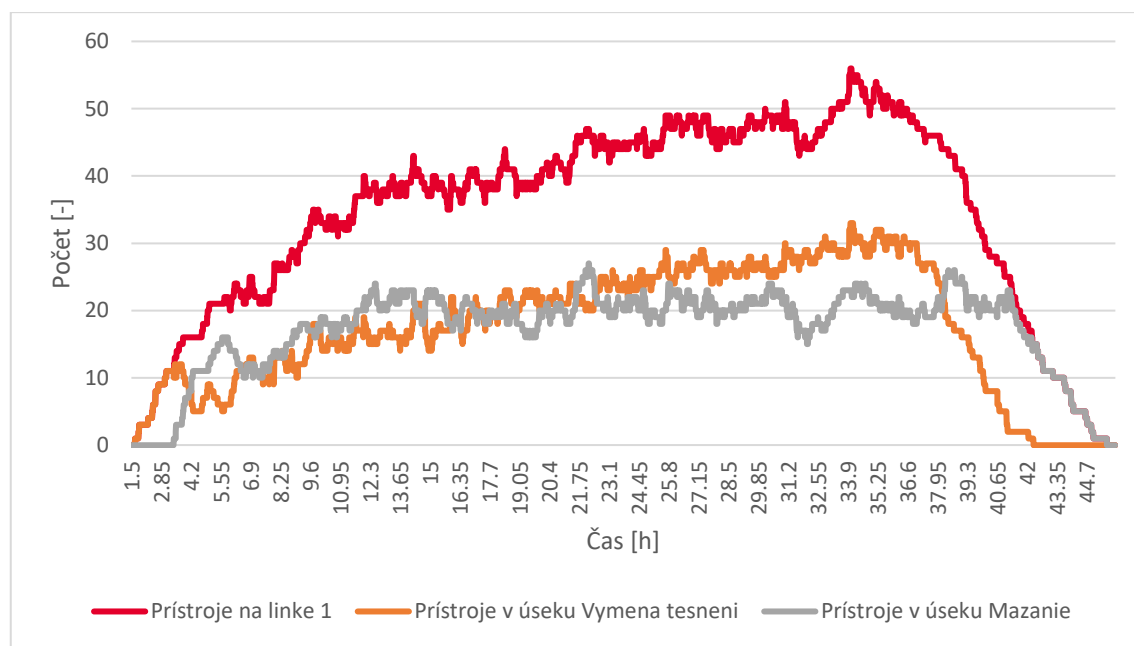
Tvary kriviek počtov prístrojov dokážu vyjadriť veľa o chode strediska. Každú krivku je možné rozdeliť na 3 oblasti. Podobný princíp rozdelenia krivky na časti platí aj pre tvary kriviek úsekov.

Prvá oblasť krivky, s prudkým rastom je oblasť, kedy sa daná linka, alebo úsek naplňa (v nasledujúcich odstavcoch je pre pokrytie ako úseku, tak aj linky použitý pojem časť strediska). Strmosť rastu tejto oblasti vyjadruje, ako rýchlo sa daná časť strediska naplní. Ideálne by mala byť táto oblasť skoková, čo by znamenalo, že všetci technici bez čakania začali pracovať. Reálne tento prípad nastane len pokiaľ je fronta tejto časti strediska adekvátne naplnená prístrojmi pred zahájením simulácie (či už vstupná fronta, alebo fronta rozpracovaných prístrojov). Pokiaľ však časť strediska po zahájení simulácie nemá dostatok prístrojov, technici musia počkať na príchod prístrojov z predchádzajúcej časti a tým sa táto časť naplňa postupne.



Po naplnení môžu nastať tri typy tvaru, alebo ich kombinácie v závislosti na prístrojoch vo fronte. Dva sa dajú popísať sklonom smernice tvaru. Čím má táto časť vyšší sklon, tým viac dochádza k hromadeniu prístrojov na tejto časti. Pokiaľ však smernica nestúpa, má táto časť strediska rovnaký interval, ako predchádzajúca časť. Pri takomto prípade nedochádza k hromadeniu, avšak vzhľadom na schopnosti technikov a frontu prístrojov môžu nastať prestoje. Tretí tvar tejto oblasti môže byť kolísavý, čo značí výrazný problém s výkonnosťou. Nie vždy je však tento problém spôsobený len zlým rozložením technikov. Kolísavý tvar môže vzniknúť aj pri niektorých prípadoch distribúcie prístrojov vo vstupnej fronte. Pre najvyššiu rýchlosť prechodu je vhodné mať úsek v saturácii bez viditeľného sklonu, avšak pre optimálnu nadväznosť a maximálne vyťaženie technikov je dobré mať mierny sklon, ktorý zabezpečí, že vo vstupnej fronte do konkrétnej časti strediska bude dostatok prístrojov a každý technik si bude môcť vybrať ten prístroj, s ktorým vie pracovať a nebude musieť čakať kým do fronty na spracovanie príde vhodný prístroj.

Posledná oblasť krivky je pri vyprázdňovaní linky. Značí, ako rýchlo prejde linka zo saturovanej do neaktívnej. V ideálnom prípade by to mala byť skoková zmena, v reálnom prípade vždy bude mať nejaký sklon, ktorý je závislý na rôznych výkonnostiach technikov. Aj pokiaľ by technici mali rovnakú výkonnosť, bol by tento čas rozdielny pre rôzne typy prístrojov. Skoková zmena by nastala, len ak by mali všetci technici rovnakú výkonnosť a zároveň by v rovnakom čase začali spracovávať rovnaké prístroje.

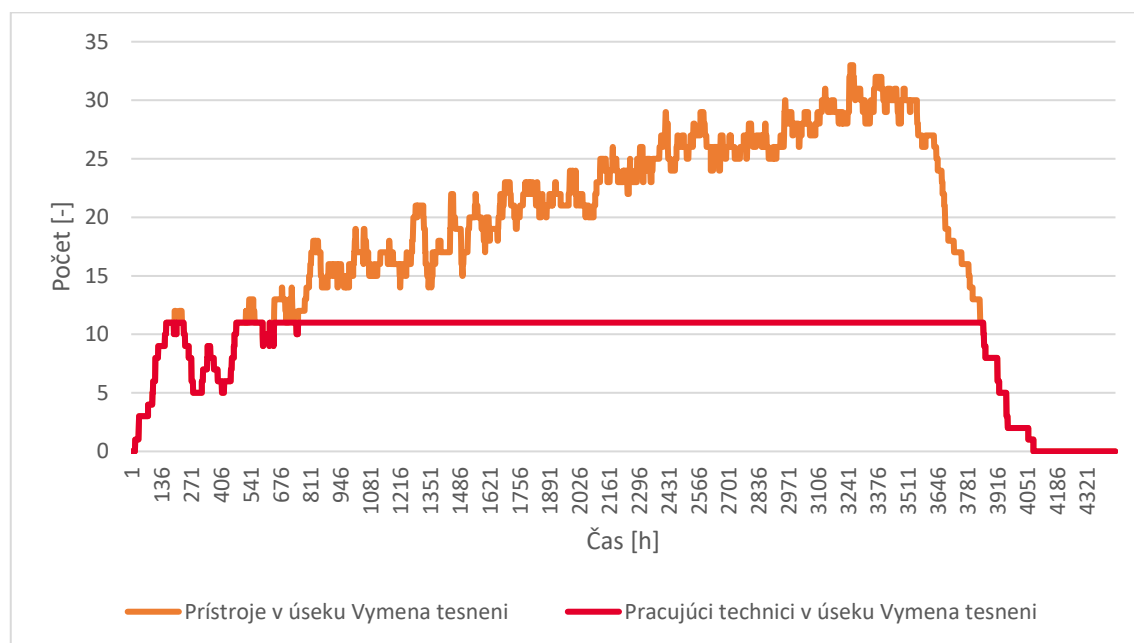


Graf 4 Detail na počty prístrojov v linke číslo 1 a jej úsekoch

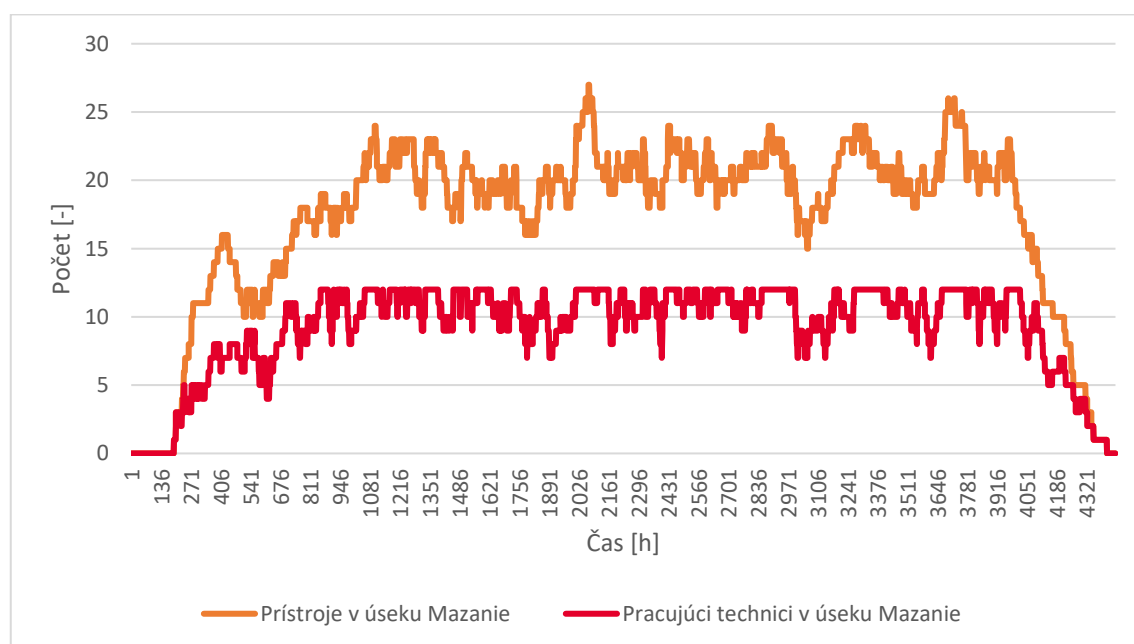
Detailnejší pohľad na prechod strediska sa dá získať vytvorením grafu prechodu prístrojov konkrétnou linkou a jej jednotlivými úsekmi (Graf 4). Saturácia linky nastáva po saturácii všetkých úsekov linky. Pre tento graf oblasť saturácie linky začína približne po 12 hodinách od začiatku simulácie. Z grafu je taktiež možné vidieť, že na úseku výmena tesnení dochádza k malému hromadeniu a úsek mazanie v saturácii spracováva prístroje tak, ako prichádzajú.

### Analýza vyťaženia technikov

Pre ešte podrobnejšiu analýzu chodu je možné vytvoriť grafy vyťaženia technikov pre samostatné úseky. Závislosťou je počet pracujúcich technikov na čase simulácie. Po preložení grafu počtu pracujúcich technikov grafom s počtom kusov prístrojov na tomto úseku (Graf 5 a Graf 6), je možné určiť, ako boli technici vyťažení a hlavne je možné sledovať prácu technikov pre konkrétnu frontu zariadení. Pomocou takéhoto grafu je totižto možné odhaliť prestoje pracovníkov z dôvodu nezaškolenia na nejaký typ prístroja, na čo sa príde až pri nejakej konkrétnej fronte zákaziek.



Graf 5 Vyťaženie technikov na úseku pre výmenu tesnení



Graf 6 Vyťaženie technikov na úseku pre mazanie častí prístroja

Počet aktívnych technikov by mal byť v ideálnom prípade vždy maximálny. V prípade, že je na úseku menší počet prístrojov, ako technikov by mal byť pre najvyššiu efektivitu vyťaženia technikov počet aktívnych technikov rovnaký, ako počet týchto prístrojov. Túto závislosť je možné vidieť na úseku pre výmenu tesnení (Graf 5). Tento úsek má v tejto simulácii všetky predpoklady k maximálnemu vyťaženiu technikov. Úsek nemá žiadne automatizované kroky, taktiež na ňom dochádza k miernemu hromadeniu, frontu prístrojov tvoria prístroje vyvážené k schopnostiam technikov a teda krivka počtu pracujúcich technikov je v maxime, prípadne kopíruje krivku počtu prístrojov.

V prípade, že sa krivka nenachádza v maxime, alebo nekopíruje tvar krivky počtu prístrojov dochádza na úseku k prestojom technikov. Tieto prestoje vznikajú pri dvoch hlavných prípadoch.

Za vznik prestojov v prvom prípade môže rozdelenie úseku na automatizované operácie. Pokiaľ je úsek optimalizovaný a má rovnaký interval ako iné úseky, nedochádza na ňom k hromadeniu a môže sa stať, že v nejakom kroku technik rýchlo spracuje všetky prístroje z fronty a musí chvíľu čakať na dokončenie nejakého z prístrojov na automatizovanom kroku. Toto je prípad zobrazený na grafe úseku mazanie (Graf 6). Na tomto grafe je možné vidieť, že je úsek vyvážený, nedochádza na ňom k hromadeniu, ale krivka je v oblasti, kde by mala byť stále v maxime rozostrená, pretože technici čakajú na prechody prístrojov automatizovanými krokmi. Pokiaľ sa na vstupe objaví prístroj pred tým, ako sa nejaký prístroj dokončí v automatizovanom kroku, technik ho samozrejme začne spracovávať. Taktiež v tomto prípade krivka vyťaženia technikov nekopíruje tvar krivky počtu prístrojov, pretože prístroje síce sú na úseku, lenže nie sú vo fronte na spracovanie, ale nachádzajú sa v krokoch s automatizovanými operáciami a čakajú na dokončenie tejto automatizovanej operácie.

Druhý prípad, kedy krivka počtu technikov nie je v maxime, alebo nekopíruje tvar krivky s počtom prístrojov nastáva v prípade, že je fronta zariadení zložená z prístrojov, na ktoré nie sú technici dostatočne zaškolení. V prípade, že je na úseku technik, ktorý by nevedel pracovať napríklad s prístrojom *Fixy-6-6i3817* a zároveň by bola fronta prístrojov zložená z viacerých po sebe idúcich týchto prístrojov, v grafe by bol na určitom mieste viditeľný konštantný posun. Tento posun signalizuje, že technik nepracuje a čaká na príchod vhodného prístroja, s ktorým dokáže pracovať. Počas tohto posuvu úsek pracuje so zníženým výkonom, bez jedného technika, jeho interval je teda vyšší, ako by mal reálne byť a na úseku vzniká hromadenie. V extrémnom prípade môže nastať prípad, kedy je fronta zložená len z prístrojov, s ktorými dokáže pracovať len jeden technik. V takom prípade sa úsek výrazne spomalí, vznikne na ňom značne vysoké hromadenie. Prechod prístrojov úsekom sa takto spomalí až na toľko, že dôjde k zabrzdeniu dynamiky celého strediska. Tento typ problému je však vďaka tomuto zobrazeniu ľahko detekovateľný a jeho náprava je jednoduchá. Stačí technikov na úseku zaškoliť na prístroje, ktoré vyžaduje fronta zariadení v stredisku. Zloženie prístrojov vo fronte zariadení je pevne dané a odvíja sa od aktuálne prijatých zákaziek.



## 4.2 Overovanie funkčnosti aplikácie

V predchádzajúcej podkapitole boli testované samotné algoritmy a ich funkčnosť. V tejto podkapitole je overená funkčnosť aplikácie vyskúšaním použitia funkcií aplikácie ako celku. Aplikácia bude mať pri tomto testovaní už načítané údaje o technikoch, výkonnostiach, prístrojoch, medzi – operáciách a modeli strediska s prednastavenými optimalizovanými hodnotami. Funkčnosť bude testovaná zrušením dostupnosti niektorých technikov, čo bude mať za následok rozhodenie balansu strediska a následného znovu nastolenia balansu automatickou optimalizáciou rozloženia dostupných technikov pre rýchlosť prechodu fronty prístrojov. Ku každému overovaniu budú priradené obrázky rozloženia technikov a taktiež grafy znázorňujúce prechody prístrojov strediskom vykreslené z údajov obsiahnutých v exportovaných csv súboroch.

### 4.2.1 Počiatočný model

Pre overenie funkčnosti aplikácie bolo treba najskôr získať dáta optimálneho chodu strediska. Tento optimálny chod je potom možné porovnávať s ďalšími modelmi, ktoré budú vznikať počas testovania funkčnosti aplikácie. Z toho dôvodu bol ako počiatočný model bol použitý model, v ktorom bolo na vstupe 250 prístrojov, pre ktoré boli technici optimalizovaní. Optimalizácia rýchlosti prechodu bola nastavená na 30 iterácií. Tento model bude slúžiť ako referenčný, z ktorého sa budú postupne nastavovaním parametrov vytvárať ďalšie modely. Aby bolo možné prečítať mená v rozložení technikov referenčného modelu, je toto rozloženie technikov zobrazené na dvoch samostatných obrázkoch (Obrázok 39 a Obrázok 40).

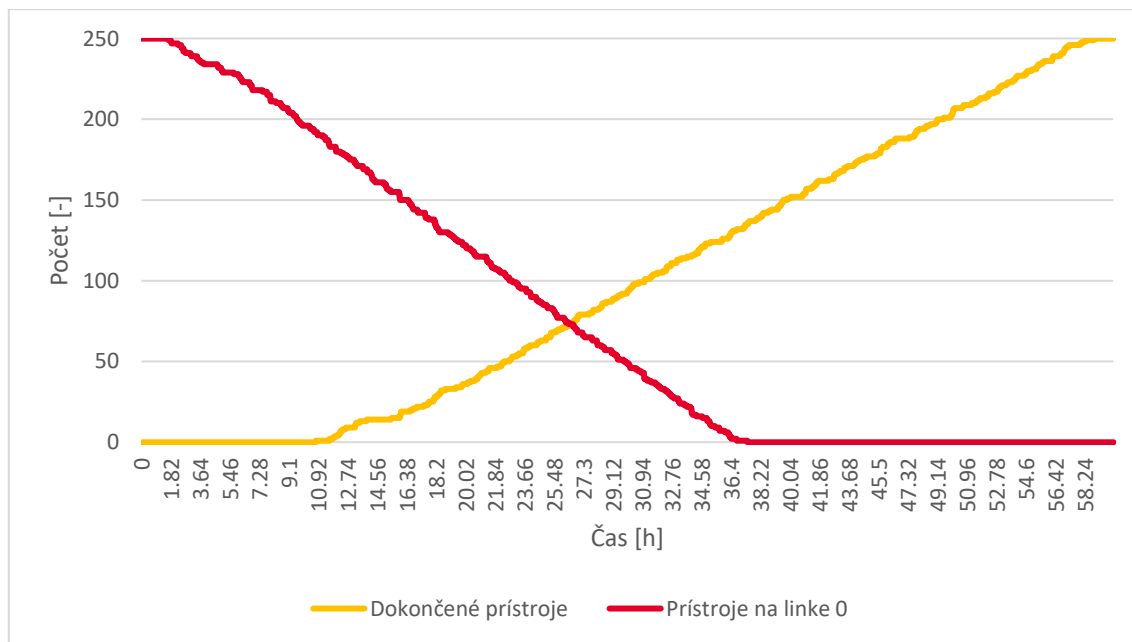
--- Linka 0 ---						
Vstupna kontrola	Vaughan Whitaker	Zachery Ferguson	Iona Lopez	Yael Montoya	Amethyst Sloan	Ethan Eaton
Diagnostika	Hall Skinner	Timon Whitney	Troy Cotton			
Rozobratie	Ivor Britt	Macy Nguyen	Carl Phillips	Forrest Bailey	Jana Obrien	Preston Morris
--- Linka 1 ---						
Vymena tesneni	Catherine Cook	Hyacinth Herrera	Christian Knight	Jeremy Farmer	Kyra Daniels	Orlando Myers
Mazanie	Chase Harding	Hector Savage	Porter McBride	Yoshi Mason	Keith Barnes	Ronan Buck
--- Linka 2 ---						
Diagnostika dosky	Anne Bentley	Jarrod Miranda	Mufutau Cervantes			
Vybratie dosky	Meredith Mckenzie	Declan Rose	Sawyer Duffy	Silas Haney	Jason Gregory	
Vymena dosky	Dora McClure	Deanna Davis	Tiger Vance			
Testovanie dosky	MacKenzie Stephens	Nasim Guerrero	Todd Rogers	Barry Marsh	Bradley Decker	
--- Linka 3 ---						
Odstranenie konektoru	Jonah Talley					
Vymena konektoru	Logan Soto					
Kontrola konektoru	Ingrid Price					
--- Linka 4 ---						
Zloženie	Leroy Sweet	Suki Bradford	Jillian Ball	Mira Valenzuela	Walter Swanson	Jarrod Wilkinson
Opt+El kontrola	Jack Carpenter	Imogene Burnett	Beverly Wolf	Julian Phillips	Emmanuel Lucas	Ann Patel
Funkcna kontrola	Abra Yang	Chandler Waller	Martina Hatfield	Adria Green	Natalie Rutledge	Katell Washington
Vystupna kontrola	Constance Parsons	Maia Finch	Micah Beasley	Leigh Cooke	Nina Aguilar	Graham Jimenez

Obrázok 39 Rozloženie technikov počiatočného modelu, prvá časť

--- Linka 0 ---						
Vstupna kontrola	Karly Mills	MacKensie Howard	Angela Sweet			
Diagnostika						
Rozobratie	Mollie Hodge	Cole Edwards	Nevada Arnold			
--- Linka 1 ---						
Vymena tesneni	Hedy Knight	Kirsten Santiago	Grady Fox	Brianna Robinson	Illiana Fleming	
Mazanie	Lucy Chandler	Brenden Mathews	Nora Rosa	Brooke Frazier	Aileen Bryan	Ira Harrell
--- Linka 2 ---						
Diagnostika dosky						
Vybratie dosky						
Vymena dosky						
Testovanie dosky						
--- Linka 3 ---						
Odstranenie konektoru						
Vymena konektoru						
Kontrola konektoru						
--- Linka 4 ---						
Zlozenie	Lynn Malone					
Opt+El kontrola	Hollie Emerson	Savannah Greer	Jaquelyn Kennedy	Gil Oconnor	Cynthia Buckner	Gareth Stout
Funkcna kontrola	Kathleen Bray					
Vystupna kontrola	Ilina Allen	Unity Garner	Jared Odonnell	Harper Watkins	Shaeleigh Justice	

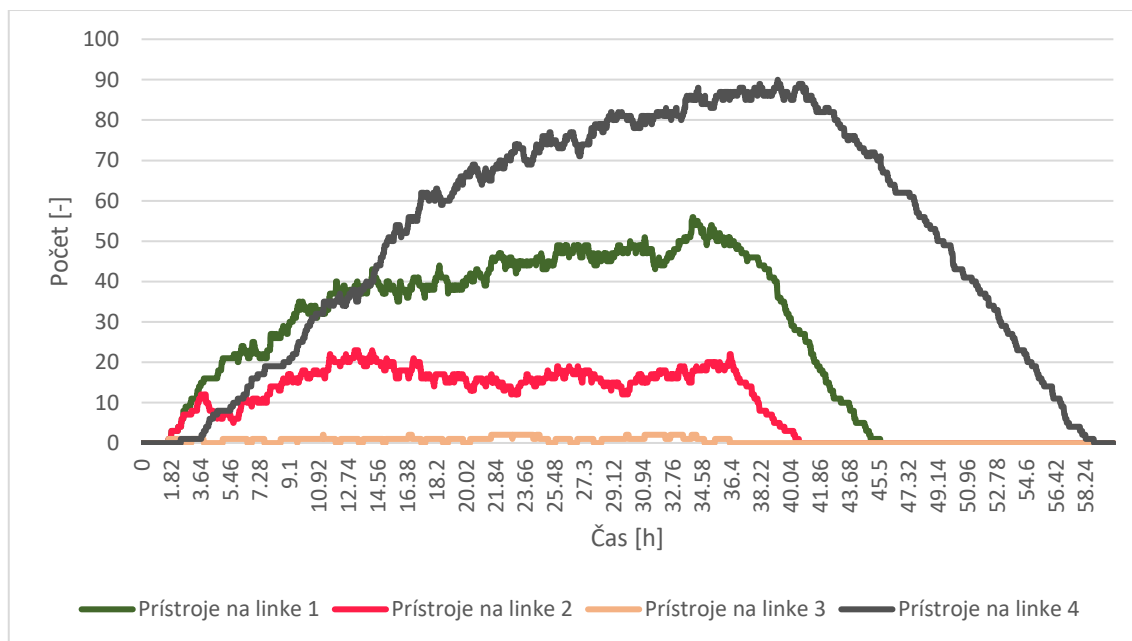
Obrázok 40 Rozloženie technikov počiatočného modelu, druhá časť

Simulácia strediska pri takomto rozložení technikov trvala 58 hodín a 47 minút. Graf s počtami prístrojov v prvej linke a počtami dokončených prístrojov (Graf 7) poukazuje na dĺžku prechodu vstupnej fronty prístrojov prípravnou linkou do strediska a na tvar krivky s počtom dokončených prístrojov. Prechod všetkých prístrojov vstupnej fronty prípravnou linkou trval približne 37,5 hodiny. Krivka dokončených prístrojov má lineárnu závislosť, bez väčších odklonov od smernice. Prechod prístrojov strediskom je teda plynulý a môžeme predpokladať, že stredisko je vyvážené.



Graf 7 Počty prístrojov v prvej linke a počty dokončených prístrojov počiatočného modelu

Sledované sú taktiež priebehy ostatných liniek v stredisku. Ich vykreslený graf je uvedený nižšie (Graf 8). V grafe sú vidieť typické prvky kriviek pri každej linke. Pri opravných linkách je možné pozorovať v prvej linke mierne hromadenie, v druhej linke mierne kolísanie a tretia linka nie je dostatočne vyťažená. Táto skutočnosť je však daná samotným modelom strediska, schopnosťami technikov a frontou zákaziek (možná úprava takéhoto typu modelu na efektívnejší spojením krátkych úsekov na linke je rozoberaná v podkapitole *Analýza vyťaženia technikov*, jedná sa však o úpravu samotného strediska). Každopádne je možné vidieť, že ako hromadenie a fluktuácia nie je moc výrazná, tak podobne aj časy ukončovania opravných liniek sú nadväzujúce, čo značí vyváženie strediska dostupnými technikmi.

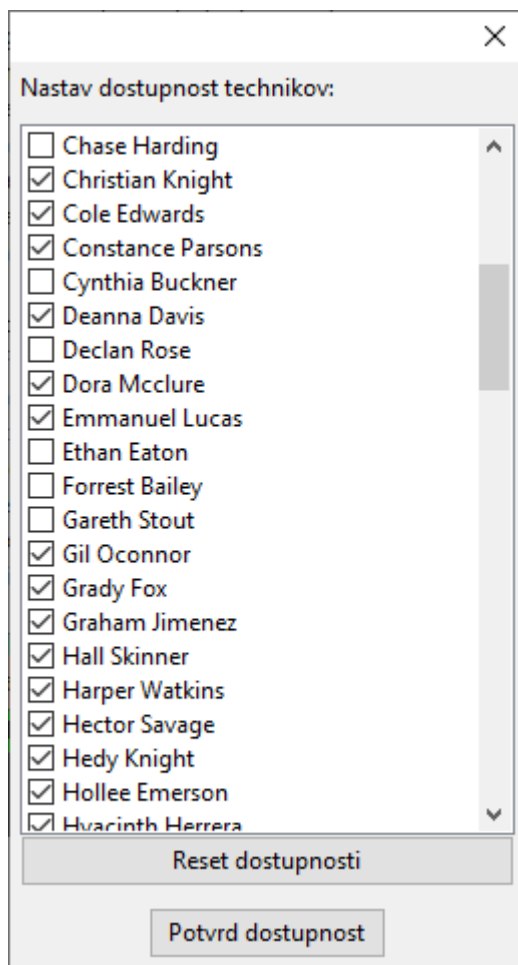


Graf 8 Prechod fronty ostatnými linkami pre počiatočný model

Pri linke číslo 4 je možné vidieť dlhý nábeh do saturácie. Tento nábeh je spôsobený čakaním jednotlivých prístrojov na príchod všetkých častí z opravných liniek. Technici môžu začať skladať len prístroje, ku ktorým majú všetky potrebné časti. Pri vyprázdňovaní dokončovacej linky je však možné vidieť podobný sklon, ako pri ostatných linkách.

### 4.2.2 Rozhodenie balansu strediska

Ďalšou fázou overovania funkčnosti aplikácie je nastavenie dostupnosti technikov. Dostupnosť technikov značí, že daný technik nemá dovolenku, prípadne voľno a je pripravený pracovať. V nastavení modelu sa pomocou zaškrŕavacích políček v dialógovom okne zruší dostupnosť nedostupným technikom. Týchto technikov aplikácia následne odstráni zo strediska a ďalej s nimi nepočíta, ani pri priradovaní technikov a ani pri optimalizovaní rozvrhnutia technikov. Obrázok dialógového okna je na obrázku nižšie (Obrázok 41).



Obrázok 41 Dialógové okno s nastavením dostupnosti technikov

Zmazanie technikov z modelu výrazne ovplyvní rozloženie výkonov na jednotlivých úsekoch a prechod fronty prístrojov strediskom sa výrazne predĺži. V krajných prípadoch, pokiaľ po odobratí technikov nebude nejaký úsek vedieť spracovať potrebné prístroje, sa po odobratí technikov stane celý model strediska neplatným. V takomto prípade je pred začatím simulácie potrebné znovu rozhodnúť technikov na jednotlivé úseky, napríklad automatizovaným priradením. Po nastavení dostupnosti v tomto príklade však ostali všetky úseky platné a mohla sa ihneď vykonať simulácia. Rozloženie technikov po nastavení dostupnosti je rozdelené na dva obrázky (Obrázok 42 a Obrázok 43).

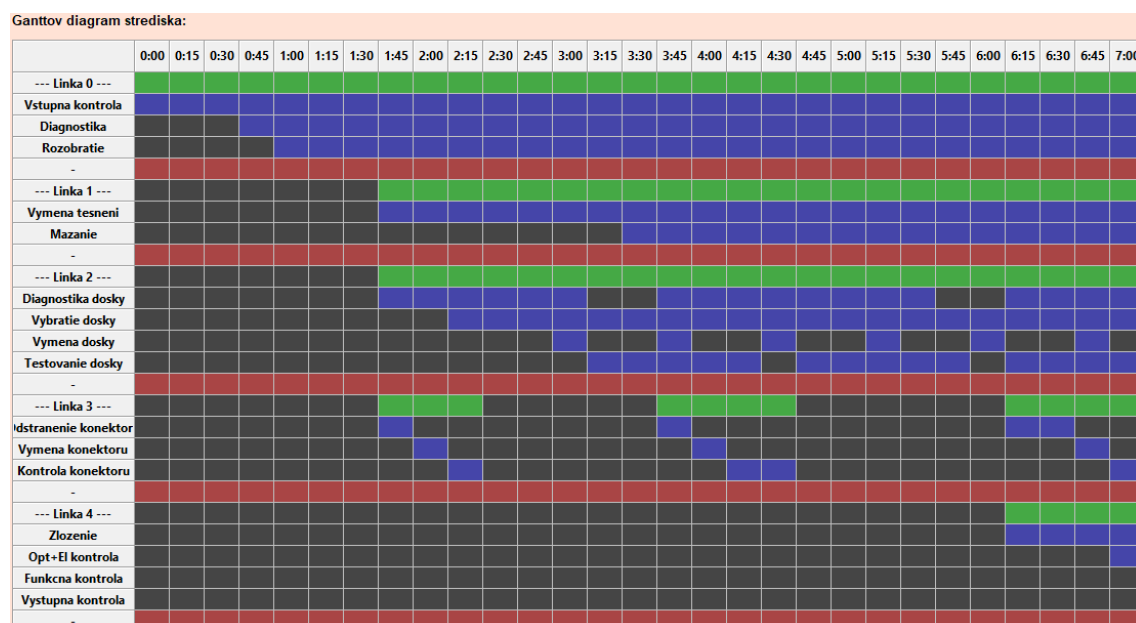
--- Linka 0 ---					
Vstupna kontrola	Vaughan Whitaker	Zachery Ferguson	Iona Lopez	Yael Montoya	Karly Mills
Diagnostika	Hall Skinner	Timon Whitney	Troy Cotton		
Rozobratie	Ivor Britt	Macy Nguyen	Carl Phillips	Jana Obrien	Preston Morris
--- Linka 1 ---					
Vymena tesneni	Hyacinth Herrera	Christian Knight	Jeremy Farmer	Kyra Daniels	Orlando Myers
Mazanie	Hector Savage	Porter McBride	Yoshi Mason	Keith Barnes	Ronan Buck
--- Linka 2 ---					
Diagnostika dosky	Anne Bentley	Jarrold Miranda	Mufutau Cervantes		
Vybratie dosky	Sawyer Duffy				
Vymena dosky	Dora McClure	Deanna Davis	Tiger Vance		
Testovanie dosky	MacKenzie Stephens	Nasim Guerrero	Todd Rogers		
--- Linka 3 ---					
Odstranenie konektoru	Jonah Talley				
Vymena konektoru	Logan Soto				
Kontrola konektoru	Ingrid Price				
--- Linka 4 ---					
Zlozenie	Leroy Sweet	Suki Bradford	Jillian Ball	Mira Valenzuela	Walter Swanson
Opt+El kontrola	Jack Carpenter	Imogene Burnett	Beverly Wolf	Julian Phillips	Emmanuel Lucas
Funkcna kontrola	Chandler Waller	Martina Hatfield	Natalie Rutledge	Katell Washington	Kathleen Bray
Vystupna kontrola	Constance Parsons	Maia Finch	Micah Beasley	Leigh Cooke	Nina Aguilar

Obrázok 42 Rozloženie technikov po odobratí nedostupných, prvá časť

--- Linka 0 ---						
Vstupna kontrola	MacKensie Howard	Angela Sweet				
Diagnostika						
Rozobratie	Mollie Hodge	Cole Edwards	Nevada Arnold			
--- Linka 1 ---						
Vymena tesneni	Hedy Knight	Kirsten Santiago	Grady Fox	Illiana Fleming		
Mazanie	Lucy Chandler	Brenden Mathews	Nora Rosa	Brooke Frazier	Ira Harrell	
--- Linka 2 ---						
Diagnostika dosky						
Vybratie dosky						
Vymena dosky						
Testovanie dosky						
--- Linka 3 ---						
Odstranenie konektoru						
Vymena konektoru						
Kontrola konektoru						
--- Linka 4 ---						
Zlozenie	Jarrold Wilkinson	Lynn Malone				
Opt+El kontrola	Hollee Emerson	Savannah Greer	Jaquelyn Kennedy	Gil Oconnor		
Funkcna kontrola						
Vystupna kontrola	Graham Jimenez	Ilina Allen	Unity Garner	Jared Odonnell	Harper Watkins	Shaeleigh Justice

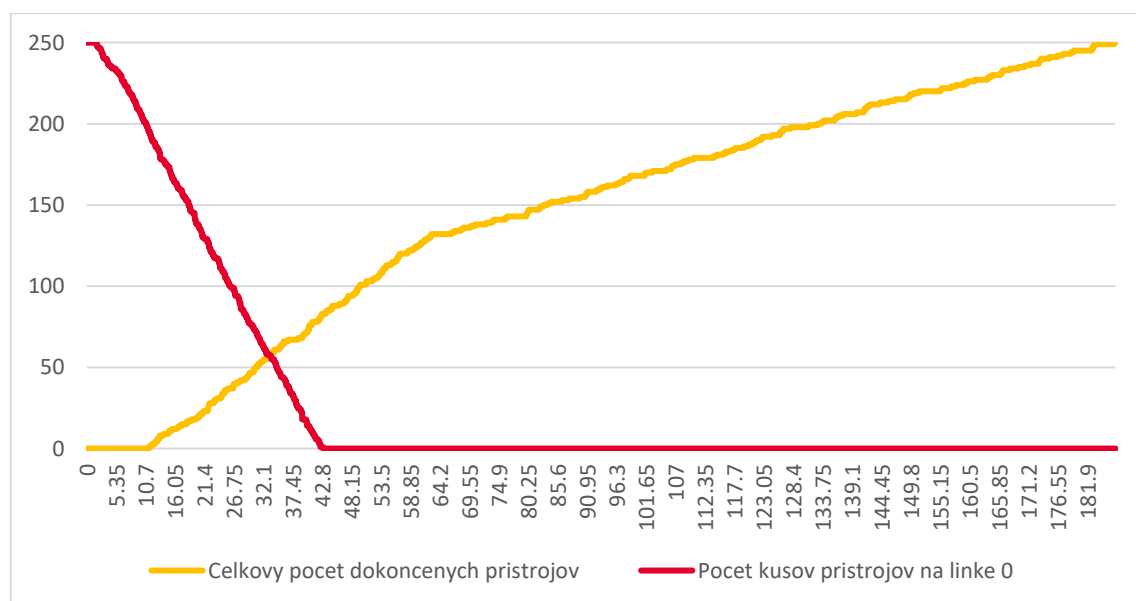
Obrázok 43 Rozloženie technikov po odobratí nedostupných, druhá časť

Simulácia pri takomto rozložení trvala až 186 hodín a 43 minút. Stredisko bolo značne nevyvážené. Túto skutočnosť naznačoval už Ganttov diagram (Obrázok 44), na ktorom je už na prvý pohľad vidieť v linke číslo 2 sivé okná neaktívnych úsekov.

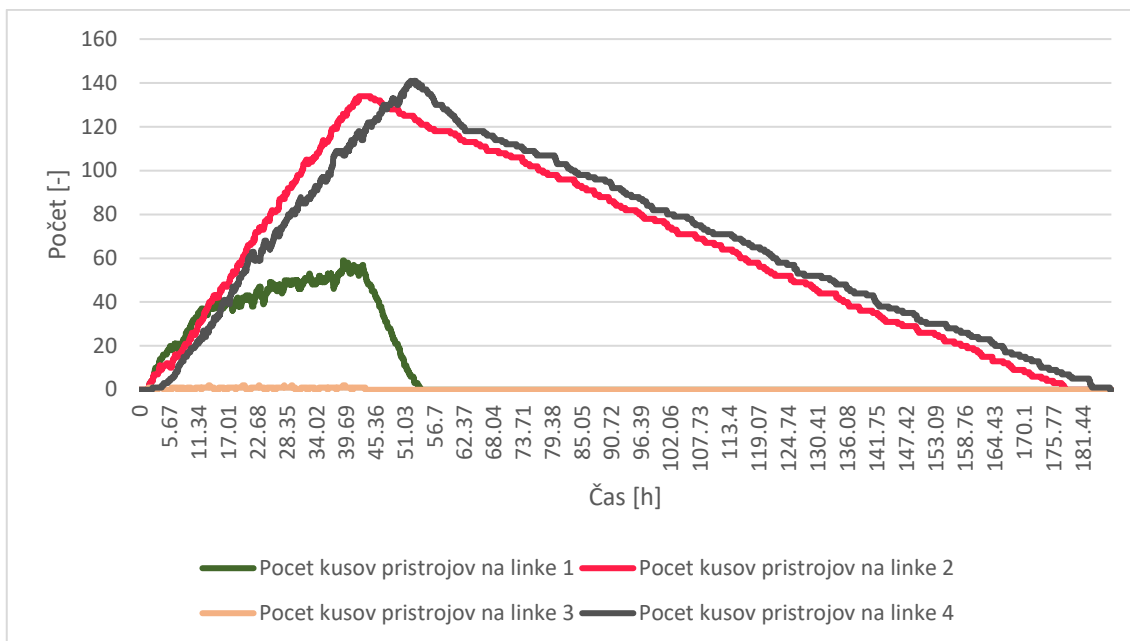


Obrázok 44 Časť Ganttovho diagramu strediska s rozhodným balansom

Graf s počtami prístrojov na prípravnej linke a počtami dokončených prístrojov (Graf 9) taktiež naznačuje značne nevyváženie strediska. Síce prechod vstupnej fronty prístrojov prípravnou linkou trval len necelých 43 hodín, krivka celkového počtu dokončených prístrojov sa viditeľne zalamuje. Toto zalomenie značí výrazne spomalenie spracovávaní prístrojov. Detailnejšiu analýzu je možné uskutočniť vykreslením grafu s počtami prístrojov na ostatných linkách strediska (Graf 10).

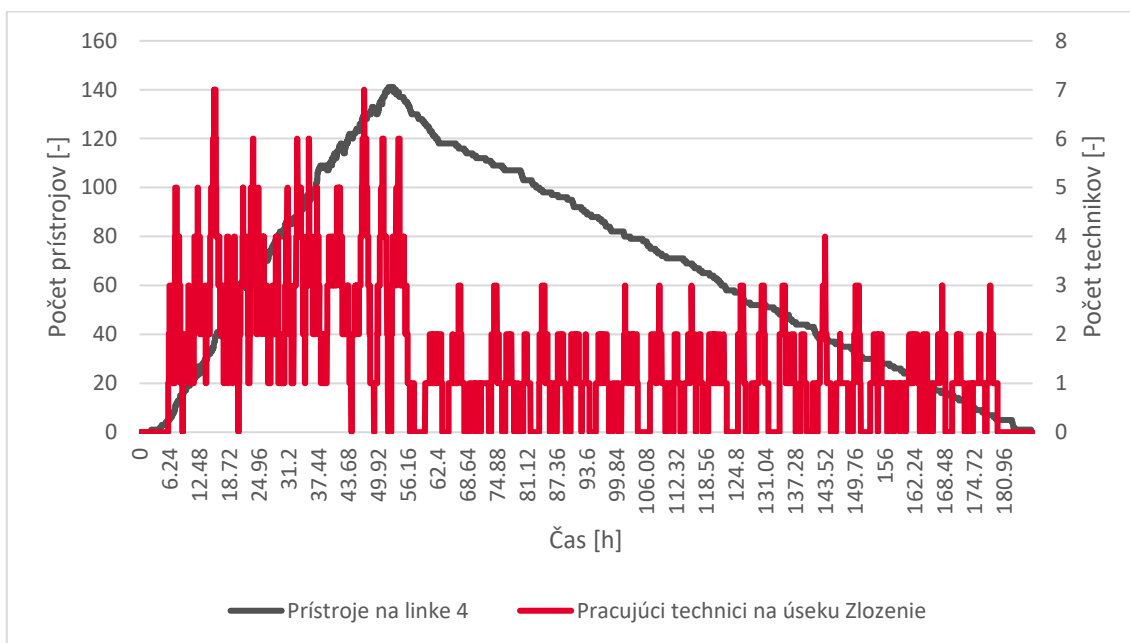


Graf 9 Počty prístrojov v prvej linke a počty dokončených prístrojov po narušení balansu



Graf 10 Prechod fronty ostatnými linkami po narušení balansu

V grafe prechodu fronty opravnými linkami a dokončovacou linkou (Graf 10) je už na prvý pohľad možné vidieť nevyváženie výkonu liniek strediska. Sklon linky číslo 2 sa výrazne odlišuje od ostatných opravných liniek, čo spôsobuje výrazné oneskorenie spúšťania dokončovacej linky (linka číslo 4). Dokončovacia linka s miernym oneskorením kopíruje tvar linky číslo 2, pretože na nej dochádza k hromadeniu prístrojov, ktoré nemajú všetky potrebné časti z linky číslo 2 pre ďalšie spracovanie. Na vstupe tejto linky dochádza k výrazným prestojom a nedostatočnému využívaniu technikov, čo je možné vidieť na grafe nižšie (Graf 11).



Graf 11 Nevyvážené vytťaženie technikov na vstupe dokončovacej linky

### 4.2.3 Optimalizácia rozloženia

Pre opätovné vyváženie strediska stačí jedným kliknutím spustiť optimalizovanie rozloženia technikov. Optimalizáciou aplikácia vytvorila rozloženie dostupných technikov, ktoré je zobrazené na obrázkoch nižšie (Obrázok 45 a Obrázok 46).

--- Linka 0 ---					
Vstupna kontrola	Vaughan Whitaker	Zachery Ferguson	Yael Montoya	Karly Mills	MacKensie Howard
Diagnostika	Dora McClure	Timon Whitney			
Rozobratie	Ivor Britt	Macy Nguyen	Carl Phillips	Preston Morris	Mollie Hodge
--- Linka 1 ---					
Vymena tesneni	Hyacinth Herrera	Christian Knight	Jeremy Farmer	Kyra Daniels	Orlando Myers
Mazanie	Hector Savage	Porter McBride	Keith Barnes	Ronan Buck	Lucy Chandler
--- Linka 2 ---					
Diagnostika dosky	Jarrod Miranda	Mufutau Cervantes	Troy Cotton		
Vybratie dosky	Sawyer Duffy	Hall Skinner	Yoshi Mason	Iona Lopez	
Vymena dosky	Deanna Davis	Tiger Vance			
Testovanie dosky	MacKenzie Stephens	Nasim Guerrero	Todd Rogers	Hedy Knight	
--- Linka 3 ---					
Odstranenie konektoru	Jack Carpenter				
Vymena konektoru	Jonah Talley				
Kontrola konektoru	Jaquelyn Kennedy				
--- Linka 4 ---					
Zloženie	Micah Beasley	Walter Swanson	Lynn Malone	Martina Hatfield	Imogene Burnett
Opt+El kontrola	Leigh Cooke	Emmanuel Lucas	Savannah Greer	Jillian Ball	Kathleen Bray
Funkcna kontrola	Jarrod Wilkinson	Katell Washington	Constance Parsons	Maia Finch	Chandler Waller
Vystupna kontrola	Nina Aguilar	Graham Jimenez	Iliana Allen	Unity Garner	Jared Odonnell

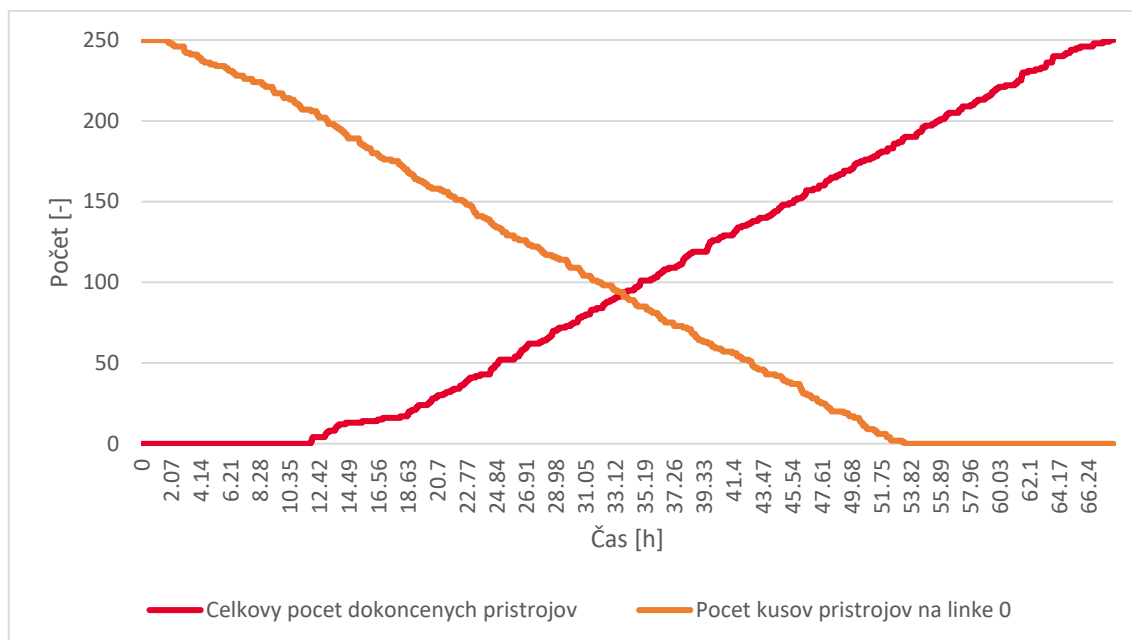
Obrázok 45 Rozloženie technikov po nasledovnej optimalizácii, prvá časť

--- Linka 0 ---					
Vstupna kontrola	Angela Sweet	Anne Bentley			
Diagnostika					
Rozobratie	Cole Edwards	Nevada Arnold			
--- Linka 1 ---					
Vymena tesneni	Kirsten Santiago	Grady Fox	Illiana Fleming	Jana Obrien	
Mazanie	Brenden Mathews	Nora Rosa	Brooke Frazier	Ira Harrell	
--- Linka 2 ---					
Diagnostika dosky					
Vybratie dosky					
Vymena dosky					
Testovanie dosky					
--- Linka 3 ---					
Odstranenie konektoru					
Vymena konektoru					
Kontrola konektoru					
--- Linka 4 ---					
Zloženie	Gil Oconnor				
Opt+El kontrola	Leroy Sweet	Logan Soto	Hollee Emerson	Ingrid Price	Mira Valenzuela
Funkcna kontrola	Natalie Rutledge				
Vystupna kontrola	Harper Watkins	Shaeleigh Justice	Suki Bradford	Julian Phillips	Beverly Wolf

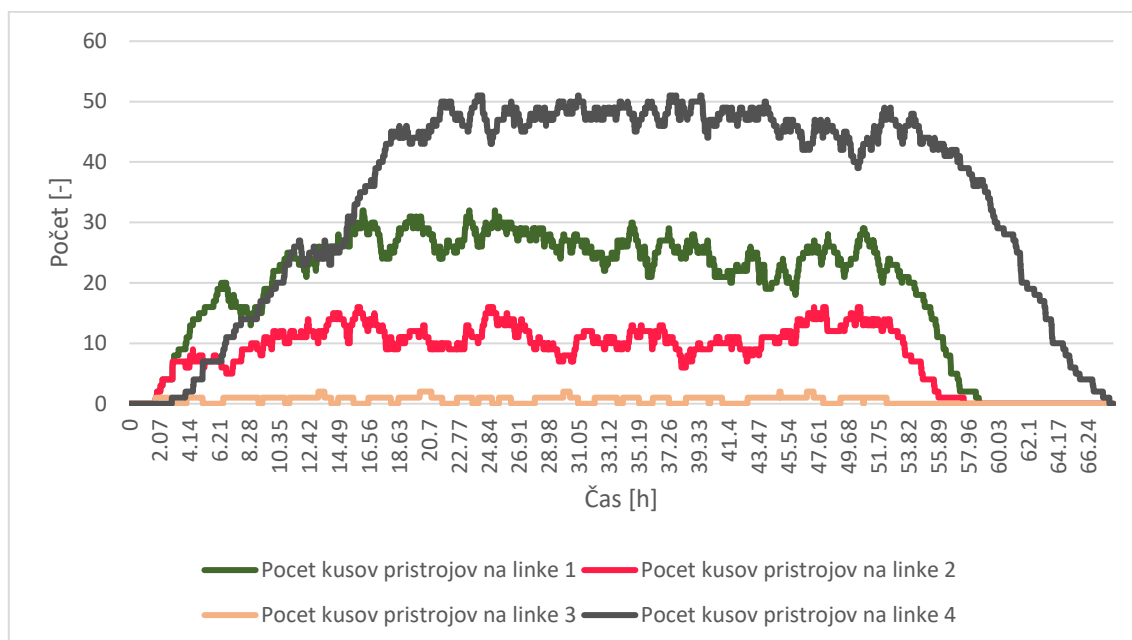
Obrázok 46 Rozloženie technikov po nasledovnej optimalizácii, druhá časť



Pri tomto rozložení technikov trval prechod fronty prístrojov strediskom len 67 hodín a 43 minút. Stredisko je opäť vyvážené. Vyváženosť je možné vidieť aj na grafoch nižšie (Graf 12 a Graf 13). Počet dokončených prístrojov v stredisku je opäť lineárny, bez výrazného zalomenia (Graf 12) a konce jednotlivých opravných liniek končia v podobné časy (Graf 13). Na žiadnej z liniek nedochádza k výraznému hromadeniu a vyprázdňovanie liniek je strmé.



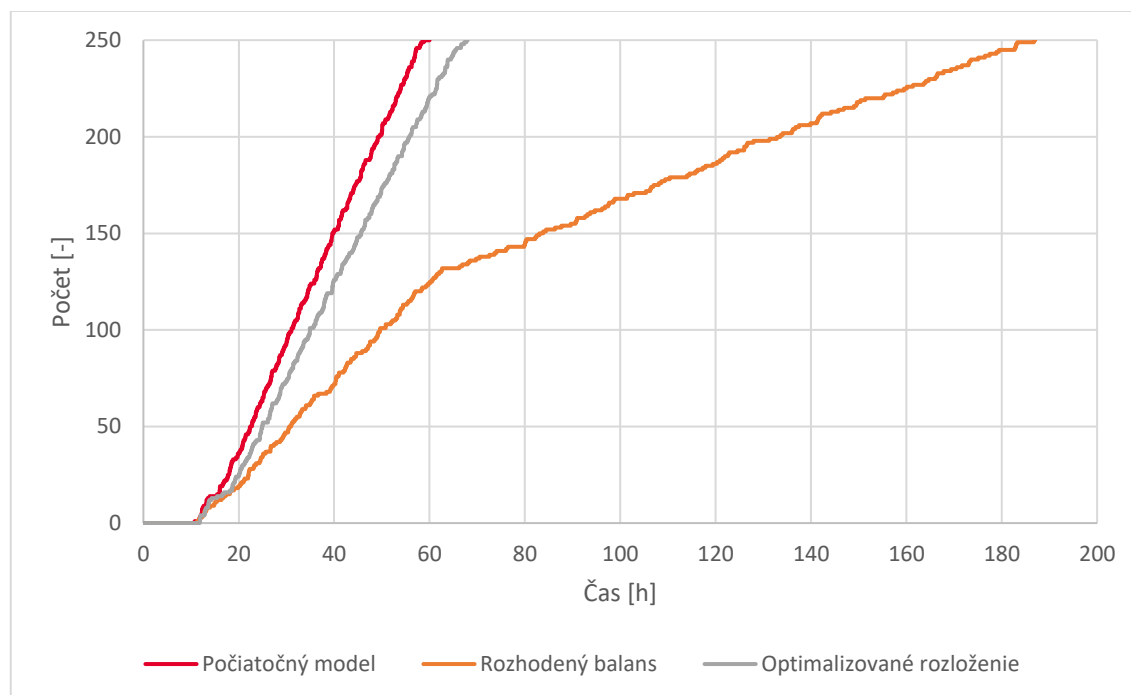
Graf 12 Počty prístrojov v prvej linke a počty dokončených prístrojov po následnej optimalizácii



Graf 13 Prechod fronty ostatnými linkami po následnej optimalizácii

#### 4.2.4 Zhrnutie overenia funkčnosti

Otestovaním aplikácie na funkčnosť celkového chodu bolo zistené, že aplikácia pracuje správne. Je možné nastavovať dostupnosť technikov a následne optimalizovať rozloženie s chýbajúcimi technikmi. Odobratie chýbajúcich technikov zo strediska výrazne rozhodilo vyváženie výkonnosti, čo bolo overené simuláciou a následne potvrdené podrobnejšou analýzou. Opätovné spustenie optimalizácie rozloženia však znovu vyvážilo rozloženie technikov a výrazne skrátilo čas prechodu fronty zákaziek. Overenie taktiež prebiehalo analýzou výstupných dát simulácie. Grafické porovnanie priebehov simulácií je možné zobrazit' pomocou grafu s porovnaním nárastu počtov dokončených prístrojov pre jednotlivé simulácie (Graf 14). Strmší nárast krivky vyjadruje vyššiu priepustnosť strediska.



Graf 14 Porovnanie dokončovania prístrojov modelmi

<b>Počiatočný model</b>	58 hodín, 47 minút
<b>Rozhodený balans</b>	186 hodín, 43 minút
<b>Optimalizované rozloženie</b>	67 hodín, 43 minút

Tabuľka 10 Prehľad dĺžok prechodov simulácií

Kvantitatívne sa výsledky jednotlivých častí overovania dajú zhrnúť do tabuľky s prehľadom dĺžok prechodov simulácií (Tabuľka 10). V tabuľke je možné vidieť, aký veľký dopad na dĺžku prechodu malo rozhodenie balansu odobratím pár technikov a do akej miery boli schopné optimalizačné algoritmy stredisko zas vyvážiť. Správnym rozložením technikov algoritmy docielili predĺženie prechodu len o necelých 10 hodín.

## 5 ZÁVĚR

Diplomová práca sa zaoberala vytvorením aplikácie pre optimalizáciu a simulovanie chodu servisného strediska. Samotné servisné stredisko má úzko špecifické potreby, ktorými sa líši od výrobného strediska. Jednotlivé zákazky na servis prístrojov vytvárajú fronty zariadení, ktoré nemajú rovnaké rozloženie prístrojov. Prístroje vo fronte bývajú rôzne a taktiež v náhodnom poradí. Z toho dôvodu bola diplomová práca rozdelená na štyri hlavné časti. Na začiatok bolo potrebné nájsť vhodné metódy, ktoré by mohli problematiku lepšie pochopiť a následne kvantifikovať. V prvej časti diplomovej práce boli popísané vhodné nástroje pre rozbor a kvantifikáciu problematiky. Druhá časť sa zaoberala analýzou a hľadaním príčin, spolu s praktickým návrhom využitia nástrojov z prvej časti. V tejto časti boli definované základné požiadavky pre funkcionálnosť aplikácie a formu výstupov.

V tretej časti diplomovej práce je navrhovaná samotná aplikácia. Od návrhu užívateľského rozhrania, až po návrh algoritmov. Pre lepšie pochopenie boli v tejto časti využívané rôzne typy štandardných UML diagramov. Návrh rozhrania sa odvíjal od potrieb pre samotnú funkcionálnosť aplikácie. K náčrtu týchto potrieb boli využívané diagramy prípadov použitia. Následne práca zhodnocuje a navrhuje využitie štandardných prvkov užívateľského rozhrania pre možnosť vykonania všetkých potrebných interakcií s užívateľom. Od prvkov potrebných pre nastavenie aplikácie a kontrolu údajov, až po prvky, ktoré sú použité pre signalizáciu a kontrolu samotného chodu aplikácie.

Algoritmy sú popísané v rôznych úrovniach abstrakcie. Pre vyššiu úroveň abstrakcie boli zvyčajne používané sekvenčné diagramy, ktoré popisovali samotnú interakciu užívateľa s aplikáciou a následnú funkčnosť pomocou volaní obsluhujúcich algoritmov. Na nižších úrovniach abstrakcie boli tieto obsluhujúce algoritmy popísané pomocou diagramov aktivít. Tento typ diagramov popisoval samotné série krokov zabalených do funkčných celkov – aktivít. Niektoré aktivity boli ešte ďalej na nižšej úrovni abstrakcie popisované ďalšími diagramami aktivít pre zobrazenie detailnejšieho chodu algoritmov, ktoré boli na vyššej úrovni abstrakcie zabalené ako jedna aktivita. Pri popisovaní algoritmov simulácie bol pre zobrazenie dátovej štruktúry simulácie využitý aj diagram triedy.

Posledná časť diplomovej práce sa zaoberala testovaním algoritmov aplikácie a overovaním funkčnosti aplikácie ako celku. Použitý model strediska bol vlastný model vytvorený pre možnosť použitia v diplomovej práci. Mená a údaje o technikoch boli vygenerované náhodne, rovnako ako boli vygenerované náhodne aj názvy prístrojov a čísla zákaziek. V diplomovej práci sa nenachádzajú reálne mená technikov, výkonnosti, zákazky a iné citlivé údaje. Testovanie funkčnosti algoritmov spočívalo v kontrole činností algoritmov pomocou analýzy správania a výstupov pri vykonávaní jednotlivých krokov, napríklad optimalizovanie, simulácia. Výstupy boli kvantifikované, najčastejšie vykreslené formou grafu. Ku grafom boli poskytnuté podrobné analýzy tvarov kriviek, pomocou ktorých je možné posudzovať výstupy algoritmov.

Overovanie funkčnosti aplikácie ako celku prebiehalo vyskúšaním typickej užívateľskej interakcie a porovnaním výsledkov. Najskôr bol nastavený počiatočný optimalizovaný model, z ktorého boli vytvorené priebehy simulácie. Výsledky boli podrobne analyzované. Čas prechodu prístrojov bol optimálny a odpovedal predpokladom.

Následne bola nastavená dostupnosť technikov, náhodne bolo vybratých pár technikov, ktorí boli označení, ako nedostupný. Pre overenie, že nedostupnosť technikov značne narušila balans strediska bola vykonaná simulácia, ktorá tento predpoklad potvrdila. Pre analýzu je okrem grafov taktiež zobrazený aj kúsok Ganttovho diagramu, v ktorom je možné vidieť dlhé okna prestojov. Celková dĺžka tejto simulácie neodpovedala optimálnemu vyváženiu rozvrhnutia technikov v stredisku, bolo možné pozorovať výrazne predĺženie času potrebného na spracovanie fronty zariadení.

Posledným krokom bolo vykonanie optimalizácie a simulácie pre potvrdenie vyváženia výkonu. Grafy potvrdili opätovné vyváženie rozloženia výkonov v stredisku. Dĺžka simulácie sa po optimalizácii rozloženia dostupných technikov výrazne skrátila. Prechod fronty prístrojov sa pomocou optimalizácie rozmiestnenia technikov výrazne zrýchlil a trval skoro len tretinový čas oproti predchádzajúcemu prípadu nevyváženého strediska.

Každý z týchto krokov bol podrobne analyzovaný a výstupy boli pre názornú ukážku výsledku riešenia problematiky dodatočne prehľadne porovnané v grafe a tabuľke v poslednej podkapitole.

Problematika optimalizácie chodu servisného strediska je pomerne zložitá a je možné ju riešiť návrhom nástroja s rôznou zložitosťou. Pre realizáciu riešenia som zvolil tvorbu zložitejšieho, ale za to komplexnejšieho riešenia, ktoré dokáže poskytnúť okrem optimalizácie a jednoduchých odhadov relatívne presnú predikciu chodu strediska v čase na základe vykonania simulácie. Všeobecne známa hláška „Simulácia je len tak dobrá, aké dobré má vstupné dáta.“ sa potvrdzuje aj v prípade navrhutej aplikácie. Čím detailnejšie budú vstupné dáta, tým detailnejšie musí byť vytvorený model strediska a tým viac sa výsledok simulácie chodu strediska priblíži realite. Komplexné riešenie taktiež poskytuje možnosti pomerne jednoducho aplikáciu rozšíriť o ďalšiu funkcionálnosť, pretože poskytuje rozsiahle univerzálne jadro aplikácie. Rozsiahlosť aplikácie ma však za následok veľa možností nastavení a s nimi spojené komplikovanejšie ovládanie aplikácie. Z toho dôvodu bola súčasťou práce taktiež tvorba užívateľského manuálu, ktorý má vo výslednej podobe 38 strán a pokrýva problematiku obsluhy aplikácie, od vytvorenia modelu, až po načítavanie výsledkov simulácie.

# LITERATURA

- [1] VOEHL, Frank. The lean six sigma black belt handbook: tools and methods for process acceleration. Boca Raton, FL: CRC Press, 2014. ISBN 978-146-6554-689.
- [2] MANDELBAUM, Jay. Value engineering synergies with lean six sigma: combining methodologies for enhanced results. Boca Raton, FL: CRC Press, c2012. ISBN 14-665-0201-0.
- [3] PYZDEK, Thomas a Paul A. KELLER. *The Six Sigma handbook: a complete guide for green belts, black belts, and managers at all levels*. 3rd ed. New York: McGraw-Hill Companies, c2010. ISBN 978-007-1623-384.
- [4] NASH, Mark A., Sheila R. POLING a Sophronia WARD. *Using lean for fast six sigma results: a synchronized approach*. New York: Productivity Press, c2006. ISBN 978-156-3273-438.
- [5] OKES, Duke. *Root cause analysis: the core of problem solving and corrective action*. Second edition. Milwaukee, Wisconsin: ASQ Quality Press, [2019]. ISBN 978-087-3899-826.
- [6] *Learning and application guide [for business graphics]*. Madison, WI: Joiner Associates, c1995. ISBN 978-188-4731-099.
- [7] HILL, Arthur V. The encyclopedia of operations management: a field manual and glossary of operations management terms and concepts. Upper Saddle River, N.J.: FT Press, c2012. ISBN 978-013-2883-702.
- [8] FRYMAN, Mark A. *Quality and process improvement*. Albany: Delmar/Thomson Learning, c2002. ISBN 978-076-6828-728.
- [9] WEALLEANS, David. *The organizational measurement manual*. Burlington, VT: Gower, c2001. ISBN 978-056-6083-495.
- [10] REID, Thomas. Sustained Leadership WBS: A Disciplined Project Approach to Building You and Your Team into Better Leaders. Morgan James Publishing, 2017. ISBN 978-168-3505-945.
- [11] BUCHTIK, Liliana. Secrets to mastering the WBS in real-world projects: the most practical approach to work breakdown structures (WBS)!. Newtown Square, Pa.: Project Management Institute, c2010. ISBN 19-355-8904-0.
- [12] *PERT guide for management use*. [Washington: For sale by the Supt. of Docs., U.S. Govt. Print. Off.], 1963.
- [13] HAUGAN, Gregory T. *Work breakdown structures for projects, programs, and enterprises*. Vienna, VA: Management Concepts, c2008. ISBN 978-156-7262-285.
- [14] TIMMRECK, Thomas C. Planning, program development, and evaluation: a handbook for health promotion, aging, and health services. 2nd ed. Sudbury, Mass.: Jones and Bartlett, c2003. ISBN 978-076-3700-621.
- [15] KIRAN, D.R. *Production Planning and Control: A Comprehensive Approach*. 1. Butterworth-Heinemann, 2019. ISBN 978-012-8189-375.
- [16] HAKSEVER, Cengiz a Barry RENDER. *Service and operations management*. New Jersey: World Scientific, [2018]. ISBN 978-981-3209-442.

- [17] PORTER, Alan L. *Forecasting and management of technology*. New York: Wiley, c1991. Wiley series in engineering and technology management. ISBN 978-047-1512-233.
- [18] JOHNS, Merida. *Creating, Coaching and Managing High-Powered Work Teams*. Morrisville, United States: Lulu.com, 2008. ISBN 978-143-0327-264.
- [19] KOENIG, Daniel T. *Manufacturing engineering: principles for optimization*. 2nd ed. Washington, D. C., c1994. ISBN 978-156-0323-013.
- [20] HELDMAN, William a Lona CRAM. *IT Project study guide*. 2nd ed. San Francisco: Sybex, c2004. ISBN 07-821-4318-0.
- [21] KLIEM, Ralph L., Irwin S. LUDIN a Ken L. ROBERTSON. *Project management methodology: a practical guide for the next millennium*. New York: Marcel Dekker, c1997. ISBN 978-082-4700-881.
- [22] KNAPP, Brent W. A Project Manager's Guide to Passing the Project Management (PMP) Exam. [www.pmexam.com](http://www.pmexam.com), 2010. ISBN 978-097-2665-674.
- [23] HAMILTON, Albert. *Managing Projects for Success: A Trilogy*. Thomas Telford, 2001. ISBN 978-072-7729-415.
- [24] BURKE, Rory a Steve BARRON. *Project management leadership: building creative teams*. Second edition. Hoboken, New Jersey: Wiley, 2014. ISBN 978-111-8825-402.
- [25] NAGARAJAN, K. *Project Management*. 2. New Age International, 2004. ISBN 978-812-2415-575.
- [26] PARMENTER, David. Key Performance Indicators (KPI): Developing, Implementing, and Using Winning KPIs. 2. John Wiley, 2010. ISBN 978-047-0593-196.
- [27] KERZNER, Harold. Project management metrics, KPIs, and dashboards: a guide to measuring and monitoring project performance. Third edition. Hoboken, New Jersey. ISBN 978-111-9427-285.
- [28] MARR, Bernard. Key Performance Indicators (KPI): The 75 measures every manager needs to know. Pearson UK, 2012. ISBN 978-027-3750-383.

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

WBS	Work Breakdown Structure, hierarchický rozklad práce.
KPI	Key Performance Indicators, klíčové ukazovatele výkonnosti
csv	Comma Separated Values, formát souboru s hodnotami oddelenými čárkou
GUI	Graphical User Interface, grafické uživatelské rozhraní
UML	Unified Modelling Language, jednotný modelovací jazyk
WIP	Work In Progress, rozpracovaná práce
MO	Medzi – operácia, použité pre skrátený názov automatizovaných činností
FIFO	First In First Out, princíp radenia fronty, kde prvá položka, ktorá vstupuje taktiež prvá aj vystúpi
CPM	Critical Path Method, metóda kritickej cesty

# SEZNAM OBRÁZKŮ

Obrázok 1 Rozdiely medzi symptómami a príčinami [5]	2
Obrázok 2 Príklady histogramu pre časové [21] a prostriedkové [25] zobrazenie	9
Obrázok 3 Diagram príčin a následkov pre plytvanie priestorom	14
Obrázok 4 Diagram príčin a následkov pre plytvanie ľudskou energiou	14
Obrázok 5 Diagram príčin a následkov pre plytvanie časom	15
Obrázok 6 Diagram prípadov použitia pre návrh aplikácie, zjednodušená verzia	18
Obrázok 7 Diagram prípadov použitia pre namodelovanie kostry strediska	20
Obrázok 8 Diagram prípadov použitia pre nastavenie a optimalizáciu parametrov modelu, zjednodušená verzia	21
Obrázok 9 Ukážka kúsku tabuľky so skrytými popismi využitej na vykreslenie mapy modelu	23
Obrázok 10 Využitie tabuľky so skrytými popismi stĺpcov pre mapu procesného modelu	23
Obrázok 11 Ukážka využitia tabuľky pre vykreslenie Ganttovho diagramu	23
Obrázok 12 Ukážka využitia stromovej štruktúry	24
Obrázok 13 Ukážka stromovej štruktúry so skrytým koreňom	24
Obrázok 14 Zjednodušený diagram aktivít algoritmu načítania technikov do stromovej štruktúry	29
Obrázok 15 Sekvenčný diagram načítania technikov od výberu súboru, po načítanie do stromovej štruktúry, zjednodušená verzia	31
Obrázok 16 Sekvenčný diagram interakcie pre nastavenie úseku, zjednodušená verzia	32
Obrázok 17 Zjednodušený sekvenčný diagram načítania prístrojov zo súboru do tabuľky	34
Obrázok 18 Diagram aktivít pre načítanie zoznamu prístrojov zo súboru	34
Obrázok 19 Zjednodušený diagram aktivít pre pridávanie dostupných spôsobilých technikov	35
Obrázok 20 Diagram aktivít výpočtu intervalu úseku, zjednodušená verzia	37
Obrázok 21 Zjednodušený diagram aktivít aktivovania automatizovaného priradovania technikov	42
Obrázok 22 Zjednodušený diagram aktivít jednej iterácie priradenia technikov	43
Obrázok 23 Zjednodušený sekvenčný diagram interakcie algoritmu optimalizácie	45
Obrázok 24 Zjednodušený sekvenčný diagram interakcií pri spustení simulácie	49
Obrázok 25 Zjednodušený diagram aktivít iterácie simulácie	50



Obrázok 26 Zjednodušenie zobrazenie dátovej štruktúry simulácie pomocou diagramu triedy	52
Obrázok 27 Zjednodušený diagram aktivít pre presunutie jedného prístroja	55
Obrázok 28 Zjednodušený diagram aktivít priradovania prístrojov nevyťažným technikom	56
Obrázok 29 Zjednodušený sekvenčný diagram načítavania výsledkov simulácie	58
Obrázok 30 Zmenšená ukážka textového histogramu v externom textovom editore	59
Obrázok 31 Zmenšený náhľad na stredisko bez technikov	62
Obrázok 32 Automatizované priradenie desiatich technikov na úseky	62
Obrázok 33 Zmenšený náhľad zoznamu po automatizovanom priradení všetkých technikov	63
Obrázok 34 Zmenšený náhľad na zoznam technikov po vykonaní optimalizácie	64
Obrázok 35 Zmenšený náhľad simulácie prechodu jedného prístroja Anie-8-ed5709 strediskom	66
Obrázok 36 Zmenšený náhľad simulácie prechodu jedného prístroja Fixy-6-6i3817 strediskom	67
Obrázok 37 Porovnanie časov z GUI aplikácie pre prístroj Anie-8-ed5709 a Fixy-6-6i3817	67
Obrázok 38 Zmenšený náhľad Ganttovho diagramu postupného spúšťania liniek a úsekov	68
Obrázok 39 Rozloženie technikov počiatočného modelu, prvá časť	73
Obrázok 40 Rozloženie technikov počiatočného modelu, druhá časť	74
Obrázok 41 Dialógové okno s nastavením dostupnosti technikov	76
Obrázok 42 Rozloženie technikov po odobratí nedostupných, prvá časť	77
Obrázok 43 Rozloženie technikov po odobratí nedostupných, druhá časť	77
Obrázok 44 Časť Ganttovho diagramu strediska s rozhodným balansom	78
Obrázok 45 Rozloženie technikov po nasledovnej optimalizácii, prvá časť	80
Obrázok 46 Rozloženie technikov po nasledovnej optimalizácii, druhá časť	80

## SEZNAM GRAFŮ

Graf 1 Histogram vytvorený z exportovaného csv pomocou externého tabuľkového procesoru	59
Graf 2 Prechod prístrojov strediskom	66
Graf 3 Celkový pohľad na prechod všetkých prístrojov strediskom	69
Graf 4 Detail na počty prístrojov v linke číslo 1 a jej úsekoch	70
Graf 5 Vyťaženie technikov na úseku pre výmenu tesnení	71
Graf 6 Vyťaženie technikov na úseku pre mazanie častí prístroja	71
Graf 7 Počty prístrojov v prvej linke a počty dokončených prístrojov počiatočného modelu	74
Graf 8 Prechod fronty ostatnými linkami pre počiatočný model	75
Graf 9 Počty prístrojov v prvej linke a počty dokončených prístrojov po narušení balansu	78
Graf 10 Prechod fronty ostatnými linkami po narušení balansu	79
Graf 11 Nevyvážené vyťaženie technikov na vstupe dokončovacej linky	79
Graf 12 Počty prístrojov v prvej linke a počty dokončených prístrojov po následnej optimalizácii	81
Graf 13 Prechod fronty ostatnými linkami po následnej optimalizácii	81
Graf 14 Porovnanie dokončovania prístrojov modelmi	82

## SEZNAM TABULEK

Tabuľka 1 Typy plytvania [3]	3
Tabuľka 2 Kategórie vplyvov $5M$ and $E$ [9] [8]	4
Tabuľka 3 Príklad použitia textového procesoru na vytvorenie Ganttovho diagramu [18]	7
Tabuľka 4 Analýza symptómov	13
Tabuľka 5 Parametre vhodné pre použitie v programe získane analýzou príčin a následkov	16
Tabuľka 6 Zhrnutie požiadaviek pre nastavenie úseku	32
Tabuľka 7 Technici na úseku výmena dosky	64
Tabuľka 8 Technici na úsekoch v linke 3	64
Tabuľka 9 Technici na úseku zloženie	65
Tabuľka 10 Prehľad dĺžok prechodov simulácií	82

## A ZDROJOVÝ KÓD DÁTOVEJ ŠTRUKTÚRY SIMULÁCIE

```
struct PristrojVStredisku
{
    std::string Nazov;
    std::string Zakazka;
    int pocetKusovVOprave;
    double casDokoncenia;
    bool dokonceny;
};
struct Pristroj
{
    std::string Nazov;
    std::string Zakazka;
    double casDoDokoncenia;
    bool caka;
};
struct Krok
{
    std::string NazovKroku;
    std::vector<Pristroj> Pistroje;
    int percentoOperacie;
    bool jeMO;
    bool jeAktivny;
    bool jePosledny;
};
struct Operacia
{
    std::string NazovOperacie;
    std::string Technik;
    std::vector<Krok> Kroky;
    bool jeAktivna;
    bool jePosledna;
};
struct castLinky
{
    std::vector<Operacia> Operacie;
    std::vector<Pristroj> Pistroje;
    long pocetPristrojov;
    int maxPocetOperacii;
    bool jeAktivna;
    bool jeParalelizovatelna;
};
```

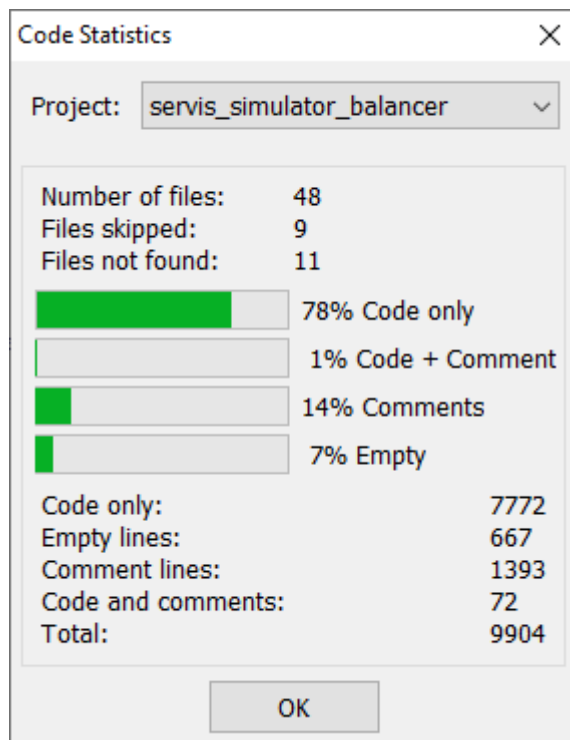
```

struct linka
{
    std::string Nazov;
    std::vector<castLinky> castiLinky;
    int percentoZoVsetkych;
    int counterPrePercenta;
    int krok;
    bool jeAktivna;
};
struct iteraciaStrediska
{
    std::vector<linka> linky;
    std::vector<PristrojVStredisku> PristrojeVStredisku;
};
struct simulacia
{
    std::string Nazov;
    std::vector<iteraciaStrediska> iteracieStrediska;
    long dlzkaSimulacie;
    int logikaRadeniaFronty;
};
std::vector<simulacia> Simulacie;

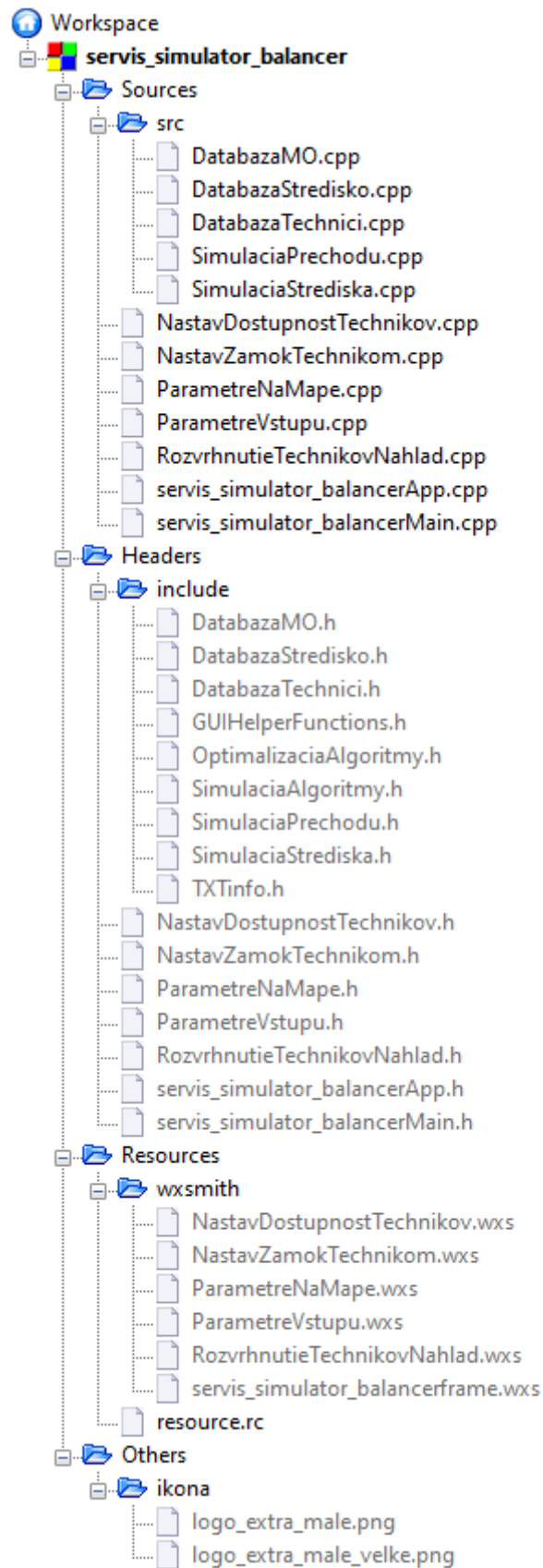
```

## B ŠTATISTIKA ZDROJOVÝCH KÓDOV

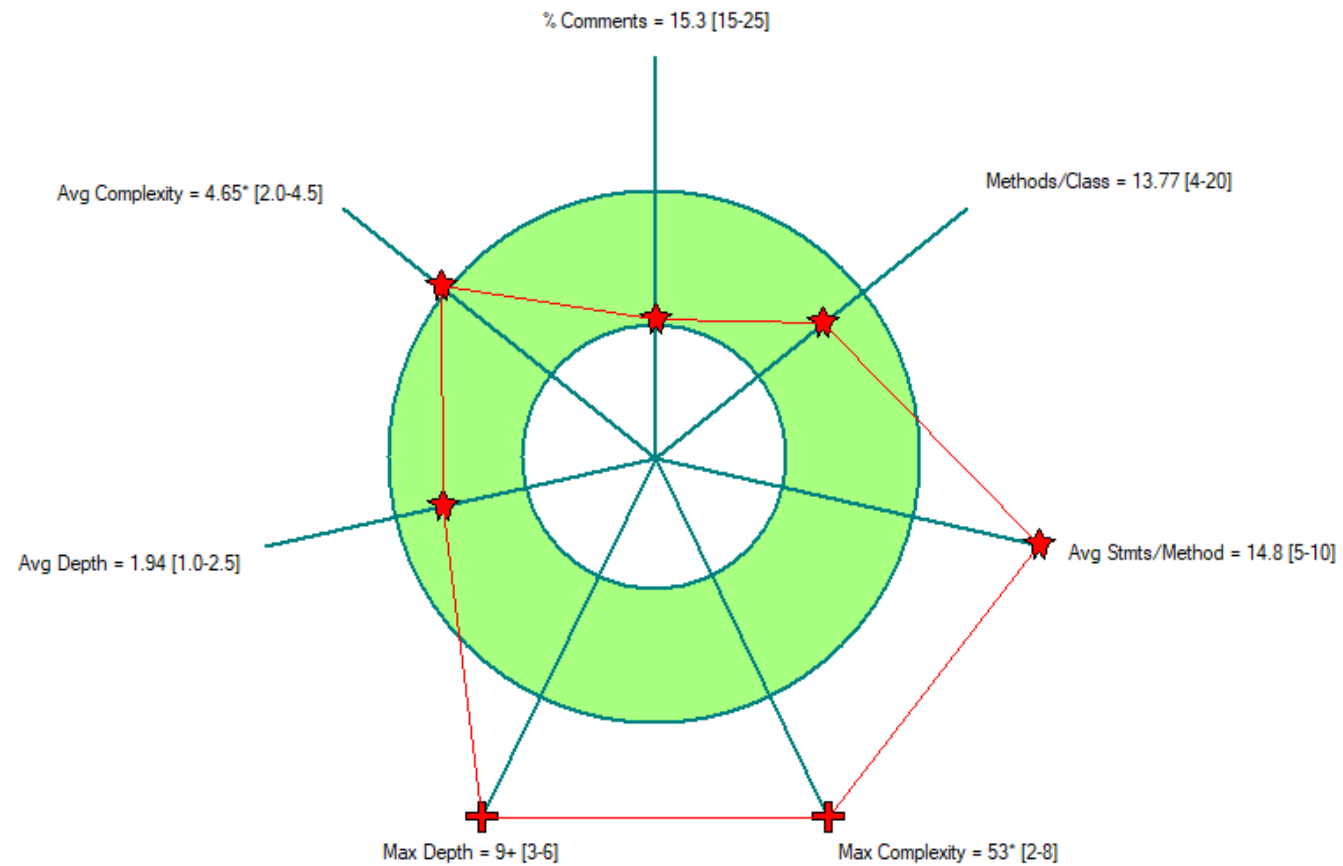
### B.1 Jednoduchá štatistika súborov projektu



## B.2 Stromová štruktúra projektu



### B.3 Kiviat diagram súborov zdrojového kódu



## B.4 Číselné štatistiky súborov zdrojového kódu

File Name	Lines	Statements	% Comments	Max Complexity	Max Depth	Avg Depth	Avg Complexity
servis_simulator_balancerMain.cpp	2,196	1,506	18.8	19*	4	0.99	2.72*
include\SimulaciaAlgoritmy.h	1,599	729	17.3	49*	9+	3.42	7.30*
include\OptimalizaciaAlgoritmy.h	1,051	535	6.2	53*	9+	3.93	12.88*
src\SimulaciaStrediska.cpp	972	281	47.4	22*	8	2.98	5.80*
ParametreNaMape.cpp	596	401	6.5	19*	7	1.32	4.29*
src\DatabazaStredisko.cpp	592	330	5.1	18*	6	2.19	2.17*
include\GUIHelperFunctions.h	538	281	6.9	15*	7	2.62	5.88*
servis_simulator_balancerMain.h	435	398	3.7	0*	1	0.94	0.00*
src\DatabazaTechnici.cpp	393	193	1	19*	5	1.62	3.45*
src\SimulaciaPrechodu.cpp	347	187	11.2	11*	8	2.15	4.55*
RozvrhnutieTechnikovNahlad.cpp	191	102	6.3	13*	6	1.75	7.25*
ParametreNaMape.h	137	109	5.8	0*	1	0.83	0.00*
NastavDostupnostTechnikov.cpp	96	44	11.5	5*	3	1.05	3.00*
ParametreVstupu.cpp	94	58	12.8	2*	2	0.79	1.25*
include\DatabazaStredisko.h	88	50	26.1	0*	1	0.92	0.00*
src\DatabazaMO.cpp	84	44	2.4	4*	3	1.07	1.75*
NastavZamokTechnikom.cpp	83	38	12	6*	3	1.13	3.67*
include\DatabazaTechnici.h	52	33	0	0*	1	0.88	0.00*
ParametreVstupu.h	51	24	15.7	0*	1	0.63	0.00*
NastavDostupnostTechnikov.h	49	21	16.3	0*	1	0.62	0.00*
NastavZamokTechnikom.h	46	18	17.4	0*	1	0.56	0.00*
include\SimulaciaStrediska.h	44	24	0	0*	1	0.63	0.00*
RozvrhnutieTechnikovNahlad.h	44	17	18.2	0*	1	0.59	0.00*
include\SimulaciaPrechodu.h	35	20	0	0*	1	0.55	0.00*



<b>servis_simulator_balancerApp.cpp</b>	33	12	36.4	2*	2	0.83	2.00*
<b>include\DatabazaMO.h</b>	23	11	0	0*	1	0.73	0.00*
<b>servis_simulator_balancerApp.h</b>	21	4	38.1	0*	1	0.25	0.00*
<b>include\SimulacaStrediska.h</b>	16	4	0	0*	1	0.5	0.00*
<b>include\TXTinfo.h</b>	14	4	42.9	0*	0	0	0.00*
<b>src\SimulacaStrediska.cpp</b>	11	3	18.2	1*	0	0	1.00*

## B.5 Číselné štatistiky metód zoradených podľa komplexnosti

Class	Method Name	Complexity	Statements	Maximum Depth	Calls
	optimalizujPreRovnomerneVytazenie()	53*	116	15	116
	infoVytazenieStrediska()	49*	114	10	76
	optimalizujPreRychlost()	43*	91	15	99
	vypocitajCasyKoeficienty()	39*	118	7	62
	textoveHistogramy()	37*	87	7	99
SimulaciaStrediska	inicializujData()	22*	73	8	60
SimulaciaStrediska	CSVExportHistogramPristrojov()	21*	66	6	71
	PriradPristroje()	20*	51	11	32
DatabazaTechnici	loadFileTechnici()	19*	66	5	33
ParametreNaMape	ParametreNaMape()	19*	229	4	301
servis_simulator_balancerFrame	OnButtonNacitatParamMapaModClick()	19*	37	4	48
DatabazaStredisko	ulozToSubor()	18*	64	6	15
DatabazaStredisko	nacitajZoSuboru()	16*	97	5	18
SimulaciaStrediska	writeProcesnyModelToGrid()	15*	32	7	38

Ukážka prvých štrnástich metód.

## C ZOZNAM ELEKTRONICKÝCH PRÍLOH

### PRILOHY

uzivatelsky\_manual.pdf

#### data

model\_technici\_bez\_ABC\_250\_pristrojov\_na\_vstupe.modp

model\_technici\_bez\_ABC\_250\_pristrojov\_na\_vstupe\_optimalizovany.modp

model\_vsetci\_technici\_250\_pristrojov\_na\_vstupe.modp

rozvrhnutie\_technikov\_model\_technici\_bez\_ABC\_250\_pristrojov\_na\_vstupe.txt

rozvrhnutie\_technikov\_model\_technici\_bez\_ABC\_250\_pristrojov\_na\_vstupe\_optimalizovany.txt

rozvrhnutie\_technikov\_model\_vsetci\_technici\_250\_pristrojov\_na\_vstupe.txt

simulacia\_model\_technici\_bez\_ABC\_250\_pristrojov\_na\_vstupe.csv

simulacia\_model\_technici\_bez\_ABC\_250\_pristrojov\_na\_vstupe\_optimalizovany.csv

simulacia\_model\_vsetci\_technici\_250\_pristrojov\_na\_vstupe.csv

#### program

readme.txt

servis\_simulator\_balancer.exe

#### ikona

logo\_extra\_male.png

logo\_extra\_male\_velke.png